МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА
Навчально-науковий Інститут енергетичної,
інформаційної та транспортної інфраструктури
Кафедра автоматизації та комп'ютерно-інтегрованих технологій

РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНА ЗАПИСКА ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

на тему Автоматизована система контролю робочих параметрів транспортних засобів

Виконав: здобувач вищої освіти
3 курсу, групи Сінж 2023-1у
напряму підготовки (спеціальності)
174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Добров А.В.
(прізвище та ініціали)

Керівник доц. каф. АКІТ, к.т.н., Аксьонов Є. О.
(прізвище та ініціали, наук. ступ., вчене звання)

Рецензент к.т.н., Ківіренко О. Б.
(прізвище та ініціали, наук. ступ., вчене звання)

Харків – 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

Навчально-науковий Інститут енергетичної,

інформаційної та транспортної інфраструктури

Кафедра автоматизації та комп'ютерно-інтегрованих технологій

Освітньо-кваліфікаційний рівень – бакалавр

Галузі знань 17 «Електроніка, автоматизація та електронні комунікації»

Спеціальність 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

ЗАТВЕРДЖУЮ
Завідувач кафедри АКІТ
БАРАНОВ О. О.
«19» червня 2026 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Доброву Андрію Вікторовичу
(прізвище, ім'я, по батькові здобувача вищої освіти)

1. Тема роботи Автоматизована система контролю робочих параметрів транспортних засобів
Керівник роботи Аксьонов Євген Олександрович, к.т.н., доц. каф. АКІТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом університету від « 22 » травня 2026 року № 440-03
2. Строк подання роботи здобувачем вищої освіти « 15 » червня 2026 р.
3. Вихідні дані до роботи Існуючі засоби контролю робочих параметрів транспортних засобів, література із створення автоматизованих систем контролю робочих параметрів транспортних засобів.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): Вступ, аналіз стану проблеми, проєктування автоматизованої системи контролю, створення моделі і дослідження роботи системи, охорона праці, висновки, перелік використаних джерел, додатки.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): презентація.

6. Консультанти розділів проєкту (роботи)

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Аналіз стану проблеми	Аксьонов Є. О., к.т.н., доцент каф. АКІТ	22.05.2026	22.05.2026
Проєктування автоматизованої системи контролю	Аксьонов Є. О., к.т.н., доцент каф. АКІТ	29.05.2026	29.05.2026
Створення моделі і дослідження роботи системи	Аксьонов Є. О., к.т.н., доцент каф. АКІТ	29.05.2026	29.05.2026
Охорона праці	Малишева В. В., доцент каф. ОПтаБЖД	26.05.2026	26.05.2026

7. Дата видачі завдання « 22 » травня 2026 р.

Керівник Аксьонов Є. О.
(підпис)

Завдання прийняв до виконання Добров А.В.
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
1	Розробка 1 розділу «Аналіз стану проблеми»	01.06.2026	
2	Розробка 2 розділу «Проєктування автоматизованої системи контролю»	08.06.2026	
3	Розробка 3 розділу «Створення моделі і дослідження роботи системи»	15.06.2026	
4	Розробка 4 розділу «Охорона праці»	15.06.2026	
5	Нормоконтроль	17.06.2026	
6	Рецензування	20.06.2026	
7	Захист на ДЕК	25.06.2026	

Здобувач вищої освіти Добров А.В.
(підпис)

Керівник Аксьонов Є. О.
(підпис)

РЕФЕРАТ

Автоматизована система контролю робочих параметрів транспортних засобів — Добров Андрій Вікторович, дипломна робота бакалавра, Харків, Харківський національний університет міського господарства імені О. М. Бекетова, кількість сторінок 122, кількість таблиць 7, кількість рисунків 45, кількість джерел літератури 38.

Дипломна робота присвячена розробці автоматизованої системи контролю робочих параметрів транспортних засобів, яка забезпечує можливість діагностики транспортних засобів за протоколами різних поколінь.

ЦІЛЬ РОБОТИ: підвищення ефективності автоматизованого контролю робочих параметрів транспортних засобів шляхом розробки системи контролю з відкритою архітектурою, що поєднує можливості діагностики за протоколами різних поколінь.

ОБ’ЄКТ ДОСЛІДЖЕННЯ: процеси автоматизованого контролю робочих параметрів транспортних засобів.

ПРЕДМЕТ ДОСЛІДЖЕННЯ: методи, апаратні і програмні засоби реєстрації, обробки і аналізу робочих параметрів транспортних засобів.

ЗАДАЧІ ДОСЛІДЖЕННЯ: проаналізувати інформаційні потоки транспортних засобів і визначити перелік найбільш критичних параметрів, сформулювати вимоги до системи контролю, обґрунтувати вибір апаратних засобів, розробити структурну схему та реалізувати апаратну частину системи, розробити архітектуру програмного забезпечення мікроконтролера, виконати моделювання та дослідження роботи системи контролю у віртуальному середовищі на основі цифрових двійників, макетування та натурні випробування системи на реальному транспортному засобі.

МЕТОДИ ДОСЛІДЖЕННЯ: аналіз літературних джерел; вивчення існуючих аналогів і їх порівняльний аналіз; синтез, моделювання та тестування апаратних й програмних засобів; натурні випробування розробленої системи.

КЛЮЧОВІ СЛОВА: автоматизована система контролю, мікроконтролер, шина CAN, лінія K-Line, цифровий двійник, реверс-інжиніринг, енергонезалежна пам’ять (EEPROM).

ABSTRACT

Automated System for Monitoring Operating Parameters of Vehicles — Dobrov Andrii Viktorovych, Bachelor’s Thesis, Kharkiv, O. M. Beketov National University of Urban Economy in Kharkiv, number of pages 122, number of tables 7, number of figures 45, number of references 38.

The bachelor’s thesis is dedicated to the development of an automated monitoring system for vehicle operational parameters, which enables vehicle diagnostics using protocols of different generations.

AIM OF THE THESIS: To increase the efficiency of automated monitoring of vehicle operational parameters by developing a monitoring system with an open architecture that combines diagnostic capabilities using protocols of different generations.

OBJECT OF STUDY: Processes of automated monitoring of vehicle operational parameters.

SUBJECT OF STUDY: Methods, hardware, and software tools for recording, processing, and analyzing vehicle operational parameters.

OBJECTIVES: analyze information flows of vehicles and determine a list of the most critical parameters, formulate requirements for the control system, justify the choice of hardware, develop a structural diagram and implement the hardware part of the system, develop the architecture of the microcontroller software, perform modeling and research of the operation of the control system in a virtual environment based on digital twins, prototyping and full-scale testing of the system on a real vehicle.

RESEARCH METHODS: Analysis of literature sources; study of existing analogues and their comparative analysis; synthesis, modeling, and testing of hardware and software; field testing of the developed system.

KEYWORDS: automated monitoring system, microcontroller, CAN bus, K-Line, digital twin, reverse engineering, non-volatile memory (EEPROM).

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ ТА СТАНДАРТІВ

AEC-Q100 — галузевий стандарт надійності для автомобільних інтегральних схем;

CAN (Controller Area Network) — промислова/автомобільна мережа контролерів;

CRC — циклічний надлишковий код;

CS (Chip Select) — лінія вибору мікросхеми в інтерфейсі SPI;

DLC (Data Length Code) — код довжини даних в CAN-кадрі;

DTC (Diagnostic Trouble Codes) — діагностичні коди несправностей (коди помилок);

EEPROM — енергонезалежна пам’ять мікроконтролера;

GND — заземлення, «маса» бортової мережі;

GPIO — програмовані контакти загального призначення;

HEX — шістнадцятирічна система числення (формат представлення даних);

I^2C/TWI — послідовний асиметричний інтерфейс зв’язку між мікросхемами;

INT (Interrupt) — апаратне переривання мікроконтролера;

ISO 9141/ISO 14230 (KWP2000) — міжнародні стандарти фізичного та діагностичного рівнів автомобільного інтерфейсу K-Line;

ISO 11898 — міжнародний стандарт шини даних CAN;

ISO 15765 — міжнародний стандарт діагностики через шину CAN;

ISO 16750 — міжнародний стандарт, що визначає кліматичні та механічні навантаження на автомобільну електроніку;

K-Line/L-Line — автомобільні діагностичні лінії зв’язку;

MISO/MOSI/SCK — сигнальні лінії послідовного інтерфейсу SPI;

SRAM — енергонезалежна оперативна пам’ять;

OBD-II — стандартизований діагностичний роз’єм;

OLED — технологія графічних дисплеїв;

PWM — широтно-імпульсна модуляція;

Real-time — режим роботи в реальному часі;

RTC — годинник реального часу;

Rx/Tx — лінії приймання (Receive) та передачі (Transmit) послідовних інтерфейсів;

SAE J1850/SAE J1979 — стандарти діагностики автомобілів;

SPI — послідовний синхронний інтерфейс зв’язку;

TTL — транзисторно-транзисторна логіка;

TVS-діод (супресор) — напівпровідниковий діод для захисту від кидків напруги;

TWAI — двонаправлений автомобільний інтерфейс мікроконтролерів ESP32;

UART — універсальний асинхронний приймаче-передавач (послідовний інтерфейс);

VAG — концерн Volkswagen Aktiengesellschaft;

VCDS — програмно-апаратний комплекс для діагностики автомобілів групи VAG;

Wokwi — онлайн середовище моделювання електронних схем та мікроконтролерів;

АКБ — акумуляторна батарея;

АЦП — аналого-цифровий перетворювач;

БК — бортовий комп’ютер;

ДВЗ — двигун внутрішнього згоряння;

ДТП — дорожньо-транспортна пригода;

ЕБК — електронний блок керування;

КЗ — коротке замикання;

LC-фільтр — фільтр на основі котушки індуктивності та конденсатора;

МК — мікроконтролер;

ОС — операційна система;

ПК — персональний комп’ютер;

ПЗ — програмне забезпечення;

РК-дисплей — рідкокристалічний дисплей;

СК — система контролю;

ТЗ — транспортний засіб;

ШІМ — широтно-імпульсна модуляція.

ВСТУП

Актуальність теми дослідження обумовлена тим, що сучасний автомобіль є складним високоавтоматизованим об’єктом, ефективність та безпека експлуатації якого напряму залежать від технічного стану його ключових вузлів та агрегатів. Інтеграція цифрових шин даних (CAN, K-Line) дозволила об’єднати електронні блоки керування (ЕБК) у єдину мережу, проте штатні системи індикації більшості транспортних засобів (ТЗ) суттєво обмежують доступ водія до оперативної технічної інформації. Найчастіше водій отримує сповіщення лише у разі критичної відмови (індикатор Check Engine), що унеможливлює превентивну діагностику спорадичних несправностей, контроль реальної паливної ефективності та деградації систем у реальному часі.

Існуючі комерційні діагностичні сканери або орієнтовані на професійне використання з підключенням до зовнішніх ПК, або мають значні затримки при бездротовій передачі даних, що робить їх незручними для щоденного моніторингу. Таким чином, виникає гостра потреба в розробці автономної, високошвидкісної та низькобюджетної автоматизованої системи контролю, здатної інтегруватися у бортову мережу ТЗ, забезпечуючи водія повним спектром інформаційно-діагностичних даних безпосередньо під час руху.

Метою дипломного проєкту є підвищення ефективності автоматизованого контролю робочих параметрів транспортних засобів шляхом розробки системи контролю з відкритою архітектурою, що поєднує можливості діагностики за протоколами різних поколінь.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати інформаційні потоки ТЗ і визначити перелік найбільш критичних параметрів;
- сформулювати вимоги до СК, інтерфейсів зв’язку, можливостей підключення додаткових модулів, вимоги до користувацького інтерфейсу, параметрів живлення та енергоспоживання;

- обґрунтувати вибір апаратних засобів: обрати оптимальну мікроконтролерну базу, виходячи з критеріїв наявності необхідних периферійних інтерфейсів, швидкодії та енергоспоживання;
- розробити структурну схему та реалізувати апаратну частину системи;
- розробити архітектуру ПЗ МК, що забезпечує зчитування та обробку в реальному часі даних з шин CAN та K-Line, а також їх оптимальне відображення на дисплеї в різних режимах;
- виконати моделювання та дослідження роботи СК у віртуальному середовищі на основі цифрових двійників, макетування та натурні випробування системи на реальному ТЗ.

Об’єктом дослідження є процеси автоматизованого контролю робочих параметрів транспортних засобів.

Предметом дослідження є методи, апаратні і програмні засоби реєстрації, обробки і аналізу робочих параметрів транспортних засобів.

Методи дослідження: аналіз літературних джерел; вивчення існуючих аналогів і їх порівняльний аналіз; синтез, моделювання та тестування апаратних й програмних засобів; натурні випробування розробленої системи.

Практичне значення отриманих результатів: розроблено та виготовлено автономний діючий макет бортового контролера, який успішно інтегровано в автомобіль; створено унікальні поведінкові моделі кастомних чипів для віртуального середовища Wokwi, що дозволяють імітувати роботу контролера MCP2515 та лінії K-Line (ISO 14230); пристрій може бути адаптований для будь-якого автомобіля з сумісними протоколами обміну; отримані результати дослідження можуть бути використані в навчальному процесі.

РОЗДІЛ 1 АНАЛІЗ СТАНУ ПРОБЛЕМИ

1.1 Транспортні засоби і керування ними

Транспортні засоби є складними технічними системами, які поєднують у собі механічні, електричні та електронні підсистеми, а також програмне забезпечення, що взаємодіють між собою для забезпечення виконання основної функції — переміщення у просторі. Сучасний транспортний засіб являє собою багаторівневу систему, в якій кожен елемент виконує визначену функцію та впливає на загальну ефективність роботи.

1.1.1 Транспортний засіб як об’єкт керування

З точки зору теорії автоматичного керування транспортний засіб можна розглядати як об’єкт керування, стан якого визначається множиною змінних параметрів. До таких параметрів належать швидкість руху, частота обертання двигуна, температура, тиск у системах, витрата палива та інші характеристики. Усі ці параметри змінюються в часі та взаємопов’язані між собою.

Основною підсистемою транспортного засобу є ДВЗ та електродвигуни, які виконують функцію перетворення енергії палива або накопиченої електроенергії в механічну роботу. Робота ДВЗ базується на складних термодинамічних процесах, які включають впуск повітря, змішування з паливом, згоряння, розширення газів та випуск продуктів згоряння. Ефективність цих процесів залежить від точності керування та стабільності параметрів [37].

У сучасних транспортних засобах контроль цих процесів здійснюється автоматично за допомогою електронного блоку керування (ЕБК). ЕБК отримує інформацію від численних датчиків, аналізує її та формує керуючі сигнали для виконавчих механізмів. Це дозволяє забезпечити оптимальні режими роботи двигуна в різних умовах експлуатації [12]. Однак значна кількість транспортних засобів, особливо старого покоління, не оснащені сучасними електронними системами керування. У таких випадках керування здійснюється механічними засобами, що значно ускладнює процес контролю параметрів роботи двигуна.

Ефективне керування таким об’єктом ускладнюється детермінованим характером робочих процесів (наприклад, циклів згоряння палива), що потребує моніторингу в режимі реального часу. Основна проблема полягає в тому, що традиційні контури керування, закладені виробником у блок керування двигуном, орієнтовані на підтримку екологічних норм та середньостатистичних показників надійності. При цьому водій, як головна ланка системи прийняття рішень, часто залишається ізольованим від критично важливої діагностичної інформації, отримуючи лише узагальнені сигнали про несправності (наприклад, індикатор «Check Engine») [24].

Таким чином, проаналізувавши транспортний засіб як складний об’єкт автоматизації, що характеризується детермінованим характером термодинамічних процесів, слід зазначити, що обмеженість штатної діагностики та орієнтація заводських блоків керування виключно на узагальнені показники створюють дефіцит оперативної інформації для водія.

1.1.2 Види транспортних засобів і їх особливості

У контексті розробки систем контролю (СК) доцільно класифікувати ТЗ не лише за конструктивними особливостями, а передусім за архітектурою побудови їх інформаційних та керуючих систем, оскільки саме це визначає можливості доступу до параметрів та способи їх обробки. З урахуванням рівня цифровізації та організації обміну даними, ТЗ можна поділити на такі основні групи:

- ТЗ з обмеженою цифровізацією, у яких значна частина параметрів (тиск оливи, температура двигуна, температура впускного повітря) передається у вигляді аналогових сигналів безпосередньо на прилади. Керування в таких системах здійснюється переважно механічними засобами (карбюратор, механічні регулятори), а контроль стану двигуна базується на показах окремих індикаторів та непрямих ознаках. Такі ТЗ відзначаються простотою конструкції та відносною надійністю, однак мають низьку точність керування і практично не забезпечують можливості глибокої діагностики;

- частково автоматизовані ТЗ, у яких окремі функції контролю та керування реалізовані за допомогою електронних систем. У таких ТЗ вже застосовуються ЕБК для окремих вузлів, однак інформаційна система залишається фрагментованою. Частина даних доступна через цифрові інтерфейси, але повний доступ до параметрів обмежений. Це створює складність у побудові універсальних систем моніторингу;
- транспортні засоби з розвиненою мережевою архітектурою (CAN, LIN, FlexRay), у яких усі основні компоненти об’єднані в єдину цифрову мережу обміну даними. У таких системах інформація передається між електронними блоками у цифровому вигляді, що забезпечує високу швидкість та точність обміну. Особливістю цих транспортних засобів є велика кількість доступних параметрів і висока щільність інформаційного потоку, що, у свою чергу, потребує застосування спеціалізованих засобів обробки, фільтрації та візуалізації даних [7].

Дане дослідження орієнтовано в першу чергу на ТЗ, що оснащені стандартизованими інтерфейсами обміну даними (зокрема CAN та K-Line), але потребують розширення функціоналу для забезпечення превентивної діагностики вузлів [19].

1.1.3 Параметри, які необхідно контролювати

Для забезпечення ефективної та безпечної експлуатації транспортного засобу, а також реалізації автоматизованого контролю його робочого стану, доцільно виділити групи параметрів, що мають найбільшу інформативність і безпосередньо характеризують роботу двигуна та допоміжних систем:

- термодинамічні параметри: температура охолоджуючої рідини, моторної оливи та повітря на впуску. Контроль цих величин дозволяє оцінити теплову стабільність двигуна та своєчасно виявити відхилення від нормального температурного режиму. Перегрів або нерівномірний розподіл температур може призвести до деформації деталей, втрати мастильних властивостей оливи та загального зниження ресурсу двигуна;

- параметри сумішоутворення: склад паливо-повітряної суміші, який характеризується коефіцієнтом надлишку повітря (λ). Від правильності цього співвідношення залежить ефективність процесу згоряння, витрата палива та рівень шкідливих викидів. Відхилення від оптимального значення може викликати детонаційні явища, нестабільну роботу двигуна та прискорений знос елементів циліндро-поршневої групи;
- динамічні показники циклу: частота обертання колінчастого вала, кут випередження запалювання та тривалість упорскування палива. Контроль цих параметрів у режимі реального часу дозволяє оцінити рівномірність роботи двигуна, виявити пропуски запалювання, несправності свічок або окремих циліндрів ще до переходу системи в аварійний режим;
- параметри навантаження двигуна: тиск у впускному колекторі, положення дросельної заслінки та пов’язані з ними показники. Вони відображають поточний режим роботи двигуна та дозволяють оцінити ступінь його завантаження, що є важливим для аналізу ефективності роботи;
- електричні параметри: напруга бортової мережі та стан акумуляторної батареї. Стабільність цих показників є необхідною умовою коректної роботи електронних систем керування та вимірювальних пристроїв [12].

Усі перелічені групи параметрів безпосередньо впливають на ресурс двигуна та безпеку руху. Тому для забезпечення ефективності моніторингу стану ТЗ важливою є їх одночасна реєстрація і комплексна обробка.

1.2 Системи контролю транспортних засобів

Системи контролю транспортних засобів є складними технічними рішеннями, призначеними для збору, обробки, аналізу та відображення інформації про стан ТЗ. Їх розвиток безпосередньо пов’язаний із еволюцією автомобільної електроніки та інформаційних технологій [37].

На початкових етапах розвитку автомобілебудування контроль параметрів здійснювався за допомогою механічних та аналогових пристроїв. Такі системи відзначалися простотою та надійністю, однак мали обмежену точність і не забезпечували можливості глибокого аналізу даних.

Подальший розвиток мікроелектроніки привів до появи цифрових СК, які дозволили значно підвищити точність вимірювань та розширити функціональні можливості обробки інформації. У сучасних умовах системи контролю реалізують не лише відображення параметрів, але й виконують діагностику, аналіз та частково прогнозування технічного стану транспортного засобу [12].

1.2.1 Види систем контролю

Сучасні системи контролю транспортних засобів можна класифікувати за функціональним призначенням, архітектурою побудови та способом взаємодії з користувачем. За функціональним призначенням виділяють:

- інформаційно-вимірювальні системи, які призначені для збору та відображення первинної інформації про стан агрегатів та виконують функцію візуалізації параметрів без втручання в процес керування;
- діагностичні системи, орієнтовані на виявлення відхилень від нормальних режимів роботи, зчитування та інтерпретацію кодів несправностей (DTC), а також оцінку технічного стану окремих компонентів;
- інтегровані рішення, що поєднують функції моніторингу та активного впливу на роботу виконавчих механізмів, забезпечуючи корекцію режимів роботи в реальному часі.

За архітектурою побудови виділяють:

- централізовані системи, у яких обробка даних здійснюється одним ЕБК, що спрощує структуру, але обмежує гнучкість і масштабованість;
- децентралізовані системи, які базуються на розподіленій мережі мікроконтролерів (МК), що обмінюються даними через цифрові шини, що дозволяє підвищити надійність і забезпечити паралельну обробку даних.

За способом взаємодії з користувачем виділяють:

- вбудовані системи, інтегровані в штатну панель приладів, які забезпечують відображення обмеженого набору параметрів;
- зовнішні системи моніторингу, які реалізуються у вигляді окремих пристроїв або програмних рішень і дозволяють розширити функціональні можливості контролю [37].

Серед розглянутих архітектур систем контролю суттєву перевагу має децентралізований підхід. Найвищу гнучкість у зборі та відображенні даних забезпечують системи моніторингу, реалізовані як зовнішній автономний вузол.

1.2.2 Апаратні засоби систем контролю

Ефективність автоматизованої системи контролю безпосередньо залежить від точності, швидкодії та надійності її апаратної бази. Основними компонентами таких систем (рис. 1.1) є первинні перетворювачі (датчики), обчислювальні модулі, інтерфейсні засоби обміну даними та пристрої відображення інформації [37].

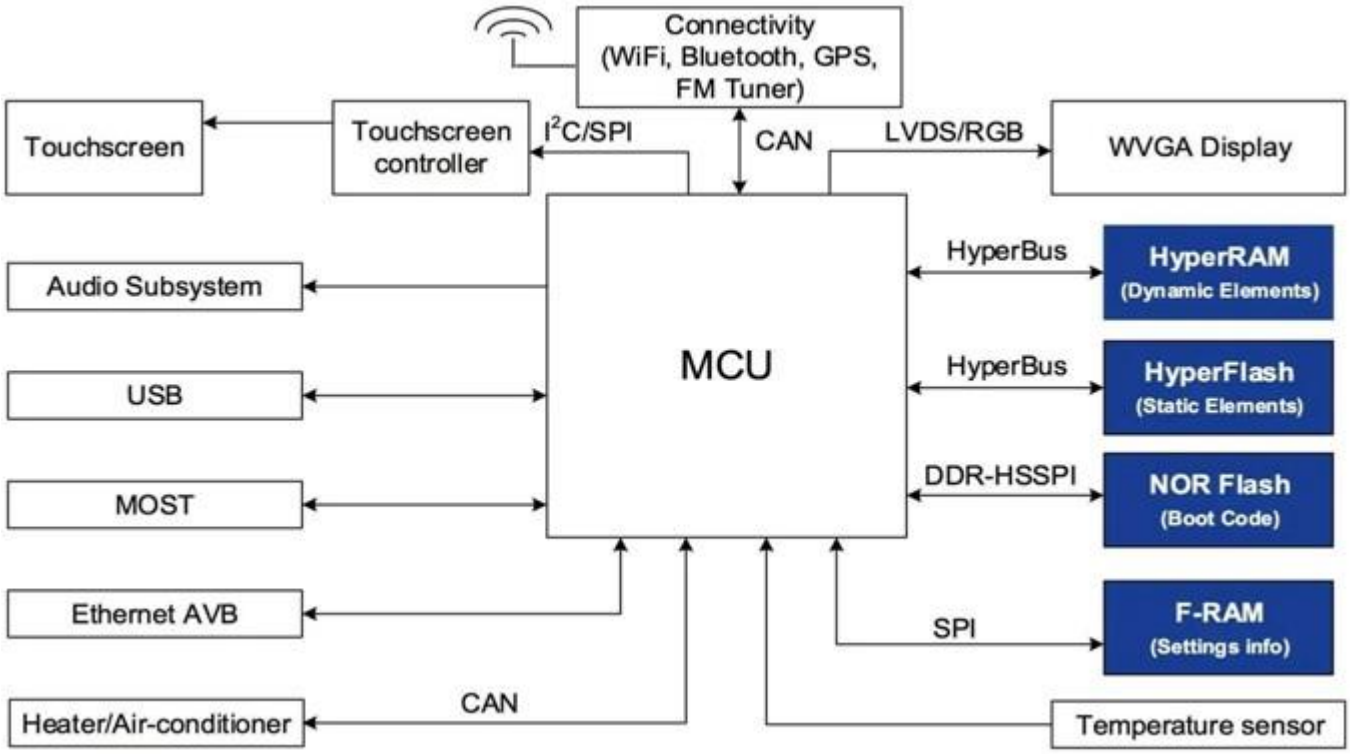


Рисунок 1.1 — Основні апаратні компоненти системи контролю ТЗ.

Датчикова апаратура є первинною ланкою системи контролю та забезпечує перетворення фізичних величин у електричні сигнали. У процесі розвитку автомобілебудування датчики пройшли шлях від простих механічних індикаторів до високоточних напівпровідникових сенсорів. У сучасних системах все частіше використовуються інтелектуальні датчики, які виконують попередню обробку сигналу безпосередньо на рівні вимірювання [12].

Для контролю параметрів роботи ДВЗ застосовуються різні типи датчиків, серед яких можна виділити наступні [12]:

- датчики Холла використовуються для визначення положення колінчастого та розподільчого валів, забезпечуючи синхронізацію роботи двигуна;
- п’єзоелектричні сенсори застосовуються для виявлення детонаційних процесів у циліндрах;
- потенціометричні та ємнісні датчики використовуються для вимірювання положення дросельної заслінки, рівня рідин та інших параметрів.

Обчислювальні модулі виконують функцію обробки сигналів і є «інтелектуальним ядром» системи. У сучасних ТЗ ці функції реалізуються ЕБК, однак у додаткових системах контролю можуть використовуватися мікроконтролерні платформи. Їх застосування дозволяє реалізувати алгоритми фільтрації шумів, які виникають у бортовій мережі внаслідок роботи генератора та системи запалювання.

Інтерфейсні модулі забезпечують передачу даних між компонентами системи. Вони можуть працювати за різними протоколами, що визначає швидкість та надійність обміну інформацією. Особливу роль відіграють пристрої відображення інформації, які забезпечують взаємодію системи з користувачем. Вони повинні забезпечувати зручне та зрозуміле представлення даних, що є важливим фактором для ефективного використання системи [37].

Таким чином, розглянувши склад типової апаратної бази систем контролю, варто зазначити переваги її реалізації на базі МК. Такий підхід дозволяє реєструвати сигнали з датчиків, застосовувати алгоритми цифрової обробки сигналів, використовувати різноманітні інтерфейси і протоколи передачі даних, забезпечувати ефективну взаємодію з користувачем.

1.2.3 Способи організації зв’язку

Обмін даними в сучасних транспортних засобах здійснюється за допомогою спеціалізованих протоколів та інтерфейсів, які забезпечують передачу інформації між окремими електронними блоками системи. Вибір конкретного способу організації зв’язку визначається вимогами до швидкості передачі, надійності, завадостійкості та функціональності системи. В автомобільному транспорті домінують мережеві стандарти, що забезпечують високу завадостійкість [7].

Послідовні інтерфейси (K-Line, L-Line) базуються на стандарті ISO 9141 та широко застосовувався у ТЗ попередніх поколінь. Передача даних здійснюється по одній лінії зв’язку, що спрощує реалізацію, але обмежує швидкість. Обмін інформацією відбувається за принципом «Master–Slave», де зовнішній пристрій ініціює запит, а ЕБК формує відповідь. Така схема забезпечує простоту взаємодії, однак призводить до затримок у передачі даних.

Послідовний інтерфейс K-Line характеризується: невисокою швидкістю передачі даних (до 10,4 кбіт/с), простотою фізичної реалізації, підтримкою діагностичних функцій. Недоліком інтерфейсу є неможливість отримання даних у реальному часі, що обмежує його використання в системах моніторингу [19].

Прикладом системи, що використовує послідовний інтерфейс, є система VCDS, яка використовується для діагностики ТЗ концерну VAG. Система дозволяє проводити: зчитування та видалення кодів несправностей (DTC), перегляд параметрів роботи систем, виконання налаштувань та адаптацій, тестування окремих вузлів. Перевагою системи є глибокий доступ до внутрішніх параметрів ТЗ. Недоліком є обмеження використання лише для певних марок транспортних засобів, а також затримки при роботі [9].

CAN (Controller Area Network) є мультимайстерною шиною обміну даними, яка широко застосовується в автомобільній техніці. Вона дозволяє одночасно підключати велику кількість електронних блоків, що обмінюються інформацією без використання центрального керуючого пристрою (рис. 1.2).

Особливістю CAN-шини є передача даних у режимі широкомовлення (broadcast), коли повідомлення доступні всім вузлам мережі. Це забезпечує високу швидкість доступу до параметрів та дозволяє отримувати інформацію практично в режимі реального часу [7]. Обмін даними з використанням CAN-шини характеризується: швидкістю передачі до 1 Мбіт/с, високою завадостійкістю, автоматичним виявленням та обробкою помилок, можливістю роботи в умовах промислових перешкод.

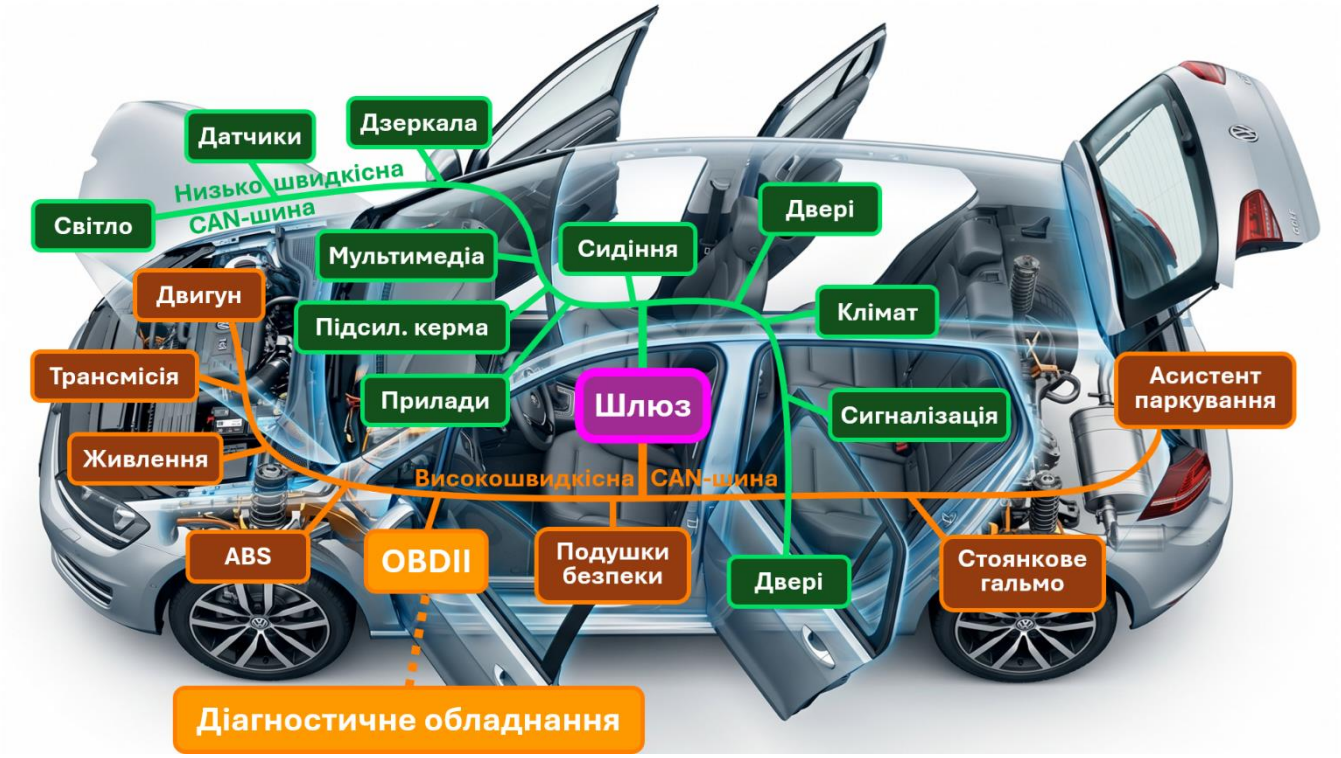


Рисунок 1.2 — Схема архітектури CAN-шини транспортного засобу.

Перевагами CAN є висока надійність, швидкість та можливість масштабування системи. Недоліком є складність структури протоколу та необхідність попереднього знання форматів повідомлень для їх інтерпретації.

Прикладом системи, що використовує вказаний інтерфейс, є CAN-аналізатори (Peak CAN, USB-CAN), які використовуються для підключення до бортової мережі ТЗ з метою зчитування та аналізу переданих повідомлень (рис. 1.3).

Рисунок 1.3 — CAN-аналізатор для роботи з шиною транспортного засобу.

Такі аналізатори дозволяють проводити: моніторинг усіх повідомлень у CAN-мережі, декодування кадрів даних, запис та подальший аналіз трафіку, інтеграцію з програмним забезпеченням для діагностики. Їхньою перевагою є можливість отримання повного доступу до інформаційного потоку, а недоліком — необхідність спеціалізованого ПЗ та знання структури CAN-повідомлень [7].

Бездротові інтерфейси, такі як Bluetooth та Wi-Fi, використовуються переважно для передачі даних від ТЗ до зовнішніх пристроїв (смартфонів, планшетів). Вони забезпечують зручність використання та мобільність, однак не є основними каналами обміну в системах реального часу. До їх переваг належать простота підключення та відсутність дротів. Водночас недоліками є затримки передачі, залежність від якості з'єднання та можливі втрати даних, що обмежує їх застосування у критичних системах контролю [9].

Аналізуючи наведені способи організації зв'язку, можна зробити висновок, що для глибокої діагностики та отримання даних у реальному часі доцільно поєднувати послідовні інтерфейси з мультимайстерними шинами. Це дозволяє компенсувати низьку швидкість діагностичних ліній високою пропускною здатністю мережевих стандартів.

1.3 Огляд існуючих систем контролю

На сучасному етапі розвитку автомобільної електроніки ринок пропонує широкий спектр засобів моніторингу. Їх можна розділити на три великі категорії: аматорські діагностичні адаптери, стаціонарні бортові комп'ютери та професійні діагностичні комплекси. Основною метою таких систем є забезпечення доступу до інформації про стан ТЗ, що дозволяє своєчасно виявляти несправності, оптимізувати режими роботи та підвищувати ефективність експлуатації. Разом з тим, кожен тип систем має свої особливості, переваги та недоліки, що визначає доцільність їх використання у конкретних умовах.

1.3.1 Портативні діагностичні адаптери на базі стандартів OBD-II

Найбільш розповсюдженим рішенням для масового користувача є адаптери ELM327, розроблені компанією ELM Electronics, та їх аналоги (рис. 1.4), апаратна частина яких базується на МК, що виконує роль інтерпретатора. Сучасні версії адаптерів здебільшого реалізуються на базі МК PIC18F25K80 або аналогів. Пристрій підтримує значну кількість автомобільних протоколів (SAE J1850 PWM/VPW, ISO 9141-2, ISO 14230-4 KWP, ISO 15765-4 CAN) [9], перетворюючи їх у стандартні інтерфейси, що підтримуються ПК та смартфонами.

Рисунок 1.4 — Зовнішній вигляд адаптерів ELM327.

Основний недолік портативних діагностичних адаптерів полягає в архітектурі обробки даних. Для отримання інформації про оберти двигуна програма повинна надіслати АТ-команду, адаптер транслює її в шину ТЗ, чекає відповіді від ЕБК, перетворює її в ASCII-текст та передає через Bluetooth. Цей ланцюжок створює критичні затримки, через які дані на екрані водія оновлюються з частотою 1–3 Гц, що є недостатнім для аналізу перехідних процесів. Також висока латентність викликана використанням бездротових інтерфейсів та послідовністю обробки стеку протоколів. Затримка оновлення даних може сягати 200–500 мс, що унеможливлює відстеження швидких процесів, таких як миттєві пропуски запалювання.

Окрім цього, такі системи зазвичай зчитують лише «загальнодоступні» параметри, не надаючи доступу до низькорівневих даних, які необхідні для глибокого аналізу стану двигуна [9], що призводить до обмеженої інформативності.

Перевагами даного типу систем контролю є низька вартість, доступність компонентної бази, універсальність щодо підтримки стандартних протоколів діагностики (SAE J1979).

1.3.2 Стаціонарні універсальні бортові комп’ютери

До цієї категорії належать автономні пристрої (наприклад, продукція брендів Multitronics, Orion), що встановлюються в салоні ТЗ на постійній основі. Вони підключаються безпосередньо до діагностичної лінії або шини CAN [7].

Бортовий комп’ютер (БК) Multitronics VC731 (рис. 1.5) є представником складних автономних СК. Він оснащений кольоровим TFT-дисплеєм та потужним внутрішнім обчислювачем, здатний працювати з оригінальними протоколами виробників ТЗ, що дозволяє йому бачити більше параметрів, ніж звичайному OBD-

сканеру. Він має функції голосового оповіщення, ведення журналів поїздок та аналізу якості пального за непрямими ознаками (зміна тривалості впорскування). Але, попри широкий функціонал, пристрій є «чорною скринькою»: користувач обмежений набором параметрів, закладених виробником.

Рисунок 1.5 — Бортовий комп’ютер Multitronics VC731.

Перевагами даного типу адаптерів є автономність (вони не потребують обов’язкового підключення до сторонніх пристроїв, як-то ПК або смартфон), наявність спеціалізованих функцій (розрахунок витрати пального за циклом, контроль напруги АКБ, сповіщення про перегрів). Недоліками стаціонарних універсальних БК є закритість архітектури (користувач не має можливості змінити логіку роботи пристрою або додати підтримку нового, нестандартного датчика), а також проблеми з ергономікою (інтерфейси таких систем часто обмежені малими сегментними або низькоякісними дисплеями, що не дозволяє виводити графіки або складні таблиці параметрів) [25].

1.3.3 Спеціалізовані програмно-апаратні комплекси

Спеціалізовані програмно-апаратні діагностичні комплекси — це професійні інструменти, які використовуються на станціях техобслуговування та працюють безпосередньо з дилерськими протоколами обміну даними.

Програмно-апаратний комплекс VCDS є фактично промисловим стандартом для діагностики ТЗ групи Volkswagen. Інтерфейс програми VCDS наведений на рисунку 1.6 [25]. Апаратна частина комплексу представляє собою спеціалізований адаптер, що підтримує як K-Line, так і CAN-інтерфейс. Система забезпечує доступ до «Груп параметрів», де дані передаються з високою роздільною здатністю. Однак,

використання режиму «Turbo», що підвищує швидкість опитування шини, часто призводить до розривів зв’язку на нестабільних лініях. Головним стримуючим фактором для постійного контролю є те, що ПЗ працює виключно під ОС Windows, що вимагає наявності ПК в салоні ТЗ, створюючи незручності та ризики під час руху.

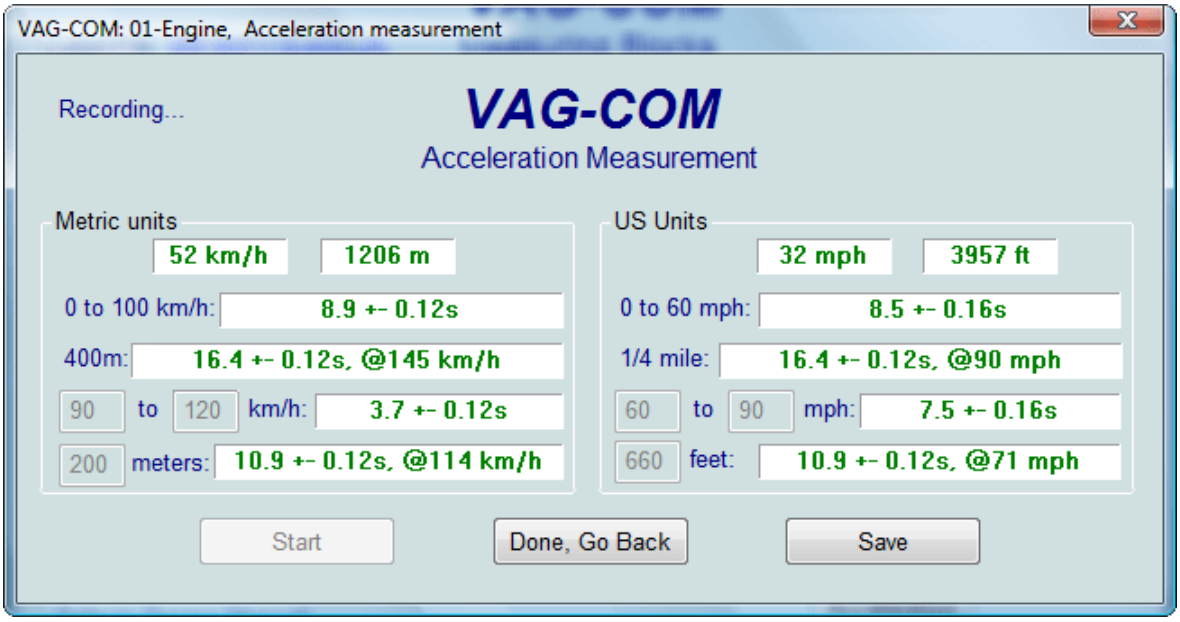


Рисунок 1.6 — Інтерфейс програми VCDS (VAG-COM) на екрані ноутбука.

Сканер Autel MaxiCOM (рис. 1.7, *а*) є професійним діагностичним пристроєм, який використовується в автосервісах. Він підтримує широкий спектр протоколів і дозволяє виконувати глибоку діагностику ТЗ [6]. До основних можливостей пристрою належать: зчитування та очищення кодів помилок, перегляд параметрів у реальному часі, тестування систем ТЗ, виконання сервісних процедур.

а) *б*)

Рисунок 1.7 — Професійні сканери Autel MaxiCOM (*а*) і Launch X431 (б).

Діагностичний комплекс Launch X431 (рис. 1.7, *б*) є ще одним прикладом професійного діагностичного комплексу. Він має подібні можливості до Autel, але відрізняється інтерфейсом та набором підтримуваних функцій.

Перевагами спеціалізованих програмно-апаратних комплексів є максимальна повнота даних, доступ до кожного датчика, що бачить ЕБК, можливість проведення тестів виконавчих механізмів. Недоліком є необхідність зовнішнього обчислювача: для роботи потрібен ноутбук або потужний планшет, що робить неможливим використання системи під час повсякденного керування авто. Також значним недоліком є вартість: професійні ліцензії та оригінальні адаптери мають високу ціну, що є недоцільним для індивідуального користувача [25].

1.3.4 Актуальність створення власного рішення

За результатами проведеного аналізу існуючих систем контролю (табл. 1.1) стає очевидним, що жодна з них не забезпечує одночасно: швидкодію рівня Real-time, доступність та можливість постійного використання в русі, відкритості коду, що дозволяє адаптувати систему під специфічні задачі (наприклад, моніторинг стану свічок запалювання через аналіз часу накопичення заряду котушок) [7].

Таблиця 1.1 — Порівняльна характеристика технічних параметрів існуючих СК.

Параметр порівняння	ELM327 (Bluetooth)	Multitronics VC731	VCDS (HEX-V2)	Розроблювана система
Швидкість оновлення даних	Низька (1-2 Гц)	Середня (3-5 Гц)	Висока (до 10 Гц)	Максимальна (Real-time)
Тип підключення	Тимчасове	Стаціонарне	Тимчасове	Стаціонарне
Модифікація ПЗ	Відсутня	Відсутня	Відсутня	Повна
Візуалізація	На смартфоні	Спрощена	Екран ПК	РК-дисплей
Читання помилок ЕБК	Тільки стандартні	Так	Так (глибоке)	Так

Сучасний рівень розвитку цифрової електроніки дозволяє реалізувати СК на базі загальнодоступних модульних апаратних платформ. Такий підхід дозволив би створити «гібридний» пристрій, який поєднає швидкість промислової шини даних з можливістю діагностики за застарілими, але все ще актуальними протоколами, заповнюючи нішу між дешевими, але повільними сканерами та дорогими

професійними комплексами [23]. Подібна система може забезпечити відкритість коду та швидкість обробки даних рівня Real-time.

Таким чином, враховуючи недоліки існуючих систем контролю і можливості сучасної цифрової електроніки, розробка автоматизованої системи контролю робочих параметрів транспортних засобів є актуальною задачею.

1.4 Постановка завдань дослідження

Як було встановлено в попередніх підрозділах, основними проблемами існуючих СК є обмежений доступ до даних, затримки у їх отриманні, необхідність використання декількох пристроїв, а також залежність від протоколів конкретного виробника ТЗ. Особливо гостро ці проблеми проявляються у випадку механічних транспортних засобів, де відсутній ЕБК або його функціональність є обмеженою.

Окрему складність становить організація доступу до різних типів даних. Дані, що характеризують роботу двигуна в режимі реального часу, передаються через CAN-шину, тоді як діагностична інформація, зокрема коди помилок, може передаватися через інші інтерфейси, такі як K-Line. Це призводить до необхідності використання різних пристроїв або складних програмно-апаратних рішень для отримання повної картини стану транспортного засобу [7].

Метою дипломного проєкту є підвищення ефективності автоматизованого контролю робочих параметрів транспортних засобів шляхом розробки системи контролю з відкритою архітектурою, що поєднує можливості діагностики за протоколами різних поколінь [23].

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати інформаційні потоки ТЗ і визначити перелік найбільш критичних параметрів;
- сформулювати вимоги до СК, інтерфейсів зв’язку, можливостей підключення додаткових модулів, вимоги до користувацького інтерфейсу, параметрів живлення та енергоспоживання;
- обґрунтувати вибір апаратних засобів: обрати оптимальну мікроконтролерну базу, виходячи з критеріїв наявності необхідних периферійних інтерфейсів, швидкодії та енергоспоживання;

- розробити структурну схему та реалізувати апаратну частину системи;
- розробити архітектуру ПЗ МК, що забезпечує зчитування та обробку в реальному часі даних з шин CAN та K-Line, а також їх оптимальне відображення на дисплеї в різних режимах;
- виконати моделювання та дослідження роботи СК у віртуальному середовищі на основі цифрових двійників, макетування та натурні випробування системи на реальному ТЗ.

Вирішення перелічених задач дозволить створити ефективну, доступну й відкриту СК, що буде позбавлена недоліків існуючих рішень і матиме перспективи подальшого розвитку.

1.5 Висновки до розділу

У першому розділі було розглянуто транспортний засіб як складний об’єкт автоматизації. Виявлено, що стандартні системи керування часто обмежують доступ водія до оперативної технічної інформації, що ускладнює превентивну діагностику.

Проведено порівняльний аналіз сучасних протоколів передачі даних. Встановлено, що для забезпечення роботи в реальному часі необхідно використовувати пряме прослуховування шини CAN, тоді як для зчитування кодів помилок доцільно використовувати лінію діагностики (K-Line).

Огляд існуючих систем контролю показав, що вони або мають великі затримки в оновленні даних, або є занадто дорогими та незручними для повсякденного використання через прив’язку до зовнішніх ПК.

Обґрунтовано актуальність розробки власного пристрою на базі мікроконтролера. Таке рішення дозволить реалізувати гібридну систему з високою швидкістю відгуку, можливістю масштабування та низькою собівартістю, що є оптимальним для індивідуального моніторингу стану технічного засобу.

РОЗДІЛ 2 ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ

2.1 Параметри, які необхідно контролювати, особливості їх контролю

Для забезпечення ефективного моніторингу технічного стану ТЗ в режимі реального часу, система повинна здійснювати безперервний збір, обробку та візуалізацію даних. Оскільки система не вводить нових датчиків у двигун, а працює з уже наявною штатною інформацією автомобіля, перелік параметрів потрібно формувати за принципом «те, що може бути стабільно одержане з CAN і K-Line без втручання в штатну електроніку». Враховуючи архітектуру сучасного автомобіля, ці дані доцільно класифікувати за функціональним призначенням: параметри двигуна, параметри акумулятора та бортової мережі, параметри допоміжних систем. Розглянемо кожну групу параметрів докладніше.

2.1.1 Параметри двигуна

ДВЗ є найскладнішим вузлом автомобіля, тому основна увага приділяється параметрам, що дозволяють оцінити його поточну динаміку та технічний стан:

- загальні експлуатаційні параметри: частота обертання колінвала, швидкість ТЗ, температура впускного повітря, положення дросельної заслінки та температура охолоджуючої рідини. Моніторинг температури є критично важливим, оскільки штатні індикатори на панелі приладів часто мають програмно згладжену шкалу, та утримують стрілку на позначці 90°C навіть при значних коливаннях;
- навантаження на двигун або тиск у впускному колекторі: розрахунковий параметр, що відображає відношення поточного циклового наповнення циліндрів повітрям до максимально можливого. Цей показник є значно інформативнішим за положення педалі акселератора, оскільки дозволяє оцінити ефективність роботи двигуна за різних умов;
- діагностика системи запалювання: виявлення пропусків запалювання як сумарно, так і по окремих циліндрах. Своєчасне виявлення деградації свічок запалювання або котушок, високовольтних дротів дозволяє запобігти пошкодженню нейтралізатора та нерівномірному зносу ДВЗ;

- параметри паливоповітряної суміші: кут випередження запалювання та дані лямбда-регулювання. Контроль напруги на датчиках кисню дозволяє оцінити якість палива та герметичність системи впуску/випуску. Оскільки ці дані мають діагностичний характер, вони можуть бути винесені на додаткові екрани інтерфейсу.

2.1.2 Параметри акумулятора та бортової мережі

Для сучасного автомобіля стабільність напруги є запорукою коректної роботи всіх електронних блоків керування.

Для розуміння процесів, що відбуваються в бортовій мережі ТЗ, важливо розділити роботу системи на кілька ключових етапів (рис. 2.1). Кожен із них характеризується своїм рівнем напруги та специфічними формами коливань:

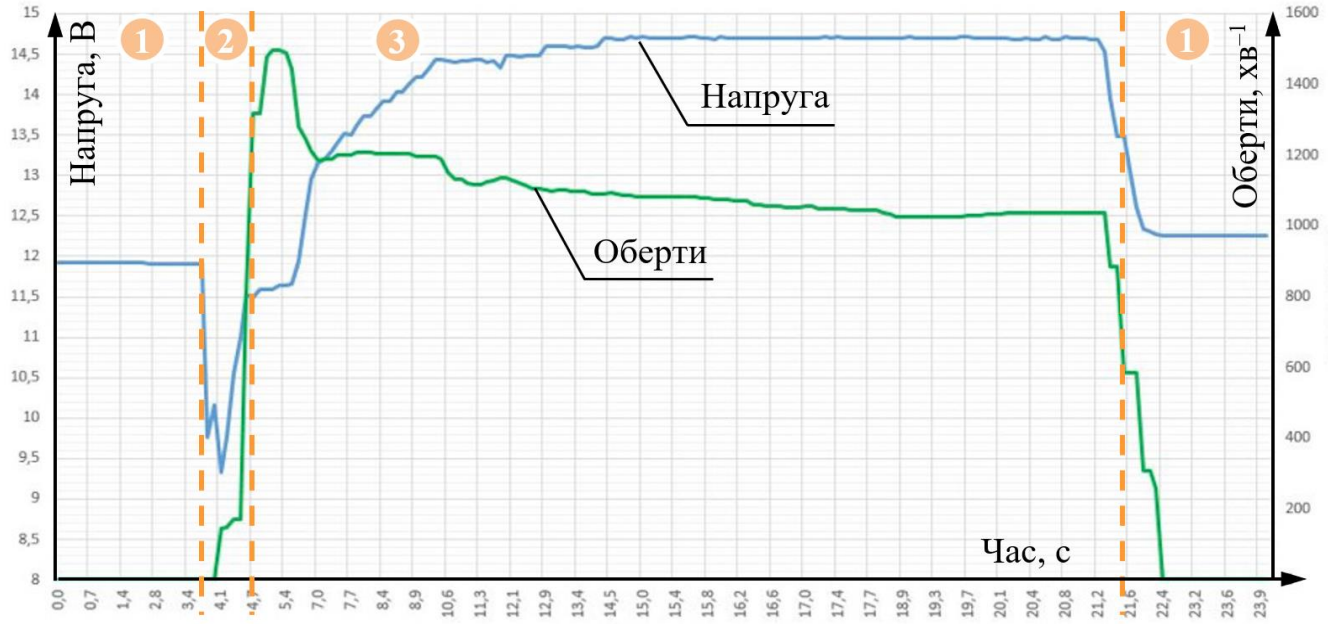


Рисунок 2.1 — Графік зміни напруги та обертів колінвалу двигуна на різних етапах роботи ТЗ.

1. Робота від акумулятора — у цьому режимі напруга стабільна і зазвичай перебуває в діапазоні 12.2 –12.7 В, вона повільно знижується в міру розряду батареї споживачами (фари, мультимедіа). Графік напруги виглядає як практично рівна горизонтальна лінія. Контроль напруги при роботі в цьому режимі дозволяє визначити рівень заряду, а контроль динаміки зміни напруги дозволяє спрогнозувати тривалість роботи від

акумулятора і сигналізувати про швидке розряджання, що може бути спричинене збільшеним споживанням, або ж поганим станом акумулятора.

2. Запуск двигуна (прокручування стартером) — це найдинамічніша ділянка графіка. У момент увімкнення стартера відбувається різке падіння напруги до 9-10 В через величезні пускові струми. Тривалість цього періоду складає від 0.5 до кількох секунд. Після початкового стрибка вниз напруга може трохи пульсувати в такт тактів стиснення в циліндрах. В цьому режимі слід контролювати, щоб рівень напруги не став критично низьким, що погано впливає на акумулятор і може призвести до його повного розряду. Також можна контролювати на скільки падає напруга відносно 1 режиму, що може сигналізувати про проблеми з двигуном, та збирати статистику напруги і тривалості запуску, відслідковувати динаміку.
3. Робота генератора (двигун запущений) — щойно двигун запускається, справний генератор піднімає напругу до 13.8 – 14.5 В. У цій фазі присутні низькоамплітудні пульсації (змінна складова), що виникають через роботу випрямного моста генератора. контроль напруги генератора. У цьому режимі корисним є контроль тривалості роботи (у випадку багатьох коротких поїздок заряд акумулятора не встигає відновлюватись) та періодичне нагадування про необхідність підзарядити акумулятор.

При вимкненні потужних споживачів (вентилятор охолодження, муфта кондиціонера) виникають короткочасні сплески зворотної полярності або позитивні імпульси амплітудою до 100 В. У цей момент генератор може видати імпульс напругою до 80–120 В тривалістю до 400 мс.

На відміну від отримання даних через запити до ЕБК (що мають певну затримку), розроблювана система повинна підтримувати пряме вимірювання напруги через вбудований АЦП мікроконтролера, що забезпечує прийнятну частоту оновлення даних та незалежність від стану діагностичної лінії. Також, для акумулятора та системи живлення доцільно контролювати температуру або оцінку стану акумулятора по наявності відповідних повідомлень у мережі.

2.1.3 Параметри допоміжних систем

Попри можливість прийому повідомлень та опитування модулів ABS, Airbag, клімат-контролю, бортової система комфорту, у даній системі має бути реалізовано принцип мінімізації запитів у шину даних для запобігання її перевантаженню.

Дані про стан систем безпеки (Airbag) або гальмівної системи (ABS) є критичними, але їх постійний моніторинг є недоцільним, оскільки при виникненні несправності відповідні індикатори активуються на штатній панелі приладів. Дані систем клімат-контролю та системі комфорту не є критичними.

Основним завданням розроблюваної системи є надання превентивної діагностичної інформації, яка відсутня на штатній панелі (наприклад, точні значення пропусків запалювання або навантаження на двигун). Опитування допоміжних модулів може бути реалізовано шляхом створення додаткових режимів відображення інтерфейсу при розширенні функціоналу пристрою під час подальшого доопрацювання проєкту. Тобто, архітектура пристрою повинна передбачати можливість введення додаткових модулів без зміни апаратної частини, а програмна частина — введення додаткових алгоритмів опитування цих модулів.

2.2 Вимоги до системи контролю

Ефективність та надійність роботи бортового пристрою моніторингу параметрів автомобіля безпосередньо залежать від чіткої ієрархії технічних та експлуатаційних вимог. Враховуючи специфіку використання системи в умовах сучасного ТЗ, вимоги до розроблюваного модуля повинні охоплювати не лише функціональні можливості обробки даних, а й стійкість до агресивного зовнішнього середовища, електричну сумісність із бортмережею та ергономічність взаємодії з водієм. Також, вимоги до системи мають враховувати розглянутий вище (в п. 2.1) перелік параметрів, пристрій повинен підтримувати усі основні інтерфейси зв’язку з автомобільною мережею, можливість підключення додаткових модулів. Користувацький інтерфейс повинен забезпечувати можливість відображення усієї необхідної інформації. Також слід контролювати споживання пристрою.

2.2.1 Загальні вимоги до автомобільних систем

Пристрій має бути реалізований у вигляді автономного модуля для встановлення на торпедо ТЗ, що висуває суворі вимоги до надійності.

Галузевий стандарт AEC-Q100 [1] офіційно задає температурні градації для інтегральних схем: Grade 0 — від -40 °C до +150 °C, Grade 1 — до +125 °C, Grade 2 — до +105 °C, Grade 3 — до +85 °C, Grade 4 — від 0 °C до +70 °C. При цьому ISO 16750-4 [18] визначає кліматичні навантаження для автомобільної електроніки залежно від місця встановлення, а системний стандарт для електричного та електронного обладнання в дорожніх транспортних засобах ISO 16750-3 [17] — механічні навантаження (вібрацію та удар). Для мікросхем AEC-Q100 також містить випробування на механічний удар та змінну вібрацію.

Тобто, компоненти та РК-дисплей, встановлені у зоні локального перегріву панелі приладів, повинні зберігати працездатність у діапазоні від -40°C до +105°C. Конструкція повинна витримувати постійні вібраційні навантаження, що досягається використанням фіксованих роз’ємів та надійним кріпленням компонентів на друкованій платі.

2.2.2 Інтерфейси передачі даних

На значній кількості автомобілів, особливо попереднього покоління, застосовується не одна загальна CAN-шина даних, а роздільні шини («двигуна» та «комфорту»), які можуть працювати на різних частотах. В шину передаються пакети з найбільш необхідною інформацією, але є частин параметрів (наприклад, кількість пропусків запалювання), отримання яких потребує виконання запитів по окремій діагностичній лінії. Тому, для забезпечення універсальності та сумісності з різними поколіннями автомобілів, система повинна підтримувати:

- шину CAN стандарту ISO 11898 [14, 15], необхідно забезпечити одночасну роботу з двома гілками шини, що працюють на різних швидкостях;
- лінію K-Line стандарту ISO 9141 [13] для діагностики за протоколами KWP1281 або KWP2000 в автомобілях попередніх років;
- фізичний рівень інтерфейсу має бути стійким до високовольтних викидів бортової мережі.

2.2.3 Можливість подальшого розширення функціональності

Проєкт передбачає відкриту архітектуру, що дозволяє нарощувати функціонал під час подальшої роботи над проєктом:

- зовнішні датчики: підтримка підключення додаткових аналогових або цифрових сенсорів (наприклад, датчика тиску турбіни або температури масла), які не передбачені заводською комплектацією;
- модулі логування: можливість встановлення SD-карти для запису параметрів під час ходових випробувань з метою подальшого аналізу;
- модуль годинника реального часу для часових міток і пробудження системи, датчик освітленості для автоматичного регулювання яскравості дисплея, модуль звукового оповіщення, блоки кнопок та енкодера для зручного користувацького введення;
- модуль зовнішнього зв’язку Bluetooth.

Це забезпечить розширюваність системи без переробки її базової апаратної платформи.

2.2.4 Вимоги до користувацького інтерфейсу

Користувацький інтерфейс розроблюваної системи повинен забезпечувати відображення основних контрольованих параметрів у реальному часі, індикацію попереджувальних і аварійних станів, а також швидке зчитування найбільш важливої для водія інформації. Відображення даних має бути організоване так, щоб пріоритетні параметри залишалися доступними без переходу у вкладені меню, а критичні повідомлення були однозначно помітними.

Окремою вимогою має бути адаптація інтерфейсу до умов освітлення в салоні автомобіля. Система повинна підтримувати автоматичне регулювання яскравості дисплея залежно від рівня зовнішньої освітленості для забезпечення читабельності вдень і зменшення засліплення водія вночі.

Система має підтримувати багаторівневу логіку керування. Навігація має виконуватись циклічним перемиканням між інформаційними сторінкам. Має бути реалізована можливість введення та зміна числових значень (наприклад, встановлення порогової температури для спрацювання тривоги, вибір швидкості

CAN-шини або зміна яскравості дисплея). Також повинна бути можливість ініціалізації діагностики шляхом вибору відповідного режиму та відправки команди на ініціалізацію читання/скидання помилок ЕБК.

Система повинна мати засоби активного сповіщення водія про вихід параметрів за межі норми: візуальне — блимання підсвічування дисплея або виведення тривожного повідомлення на екран; звукове — використання п’єзовипромінювача (буззера) для подачі звукових сигналів при помилках.

Інтерфейс має бути реалізований за модульним принципом, де додавання нового модуля (наприклад, зовнішнього датчика) автоматично створює нову інформаційну сторінку без необхідності повної перебудови логіки меню.

2.2.5 Забезпечення стабільного та якісного живлення

Пристрій, що проєктується, працюватиме в умовах значних електромагнітних перешкод та нестабільної напруги бортової мережі автомобіля (номінальна напруга 10–14.4 В для легкових ТЗ та 24–28 В для вантажних). Оскільки цифрова частина системи (мікроконтролер, модулі зв’язку та дисплей) потребує стабілізованої напруги 5 В, особливу увагу слід приділити перетворенню та захисту живлення.

Для забезпечення сумісності з різними типами транспортних засобів система повинна підтримувати широкий діапазон вхідної напруги, що дозволить нівелювати просідання напруги під час пуску двигуна та безпечно працювати при її підвищенні в бортових мережах важкої техніки.

Вхідні кола живлення повинні містити комбіновані фільтри (LC-фільтри) для придушення високочастотних завад від системи запалювання, а також захисні діоди (TVs-супресори). Це необхідно для захисту внутрішніх компонентів від короткочасних високовольтних сплесків, що можуть сягати 80–100 В.

Враховуючи роботу пристрою в закритому корпусі на торпедо автомобіля, критично важливим є мінімальне тепловиділення. Використання імпульсного (Step-Down) перетворювача замість лінійних стабілізаторів дозволяє досягти ККД понад 85%. Це гарантує відсутність перегріву компонентів навіть при значній різниці між вхідною та вихідною напругами (наприклад, при живленні від 24 В).

2.2.6 Енергоспоживання та інтелектуальні режими очікування

При розробці системи моніторингу питання енергоспоживання розглядається з урахуванням специфіки експлуатації автомобіля та необхідності збереження заряду акумуляторної батареї (АКБ) під час тривалих стоянок.

В активному режимі функціонування пристрій активується за сигналом запалювання. Оскільки основний цикл роботи припадає на час роботи двигуна (коли генератор забезпечує стабільний заряд АКБ), струм споживання в цьому режимі не є критичним обмеженням для бортової мережі.

Хоча основний режим роботи передбачає відключення при вимкненні запалювання, архітектура системи повинна закладати можливість переходу в режими глибокого сну. Це необхідно для реалізації перспективних функцій, таких як дистанційний моніторинг температури в салоні, контроль напруги АКБ у стані спокою або робота охоронних модулів. Для забезпечення гнучкості системи необхідно передбачити можливість інтеграції зовнішньої мікросхеми годинника реального часу (RTC) з функцією апаратного будильника.

Це дозволить реалізувати періодичний контроль: МК може перебувати у режимі глибокого сну з наднизьким споживанням і «прокидатися» за сигналом від RTC для швидкого зчитування параметрів і повернення у сон. Також, це дозволить уникнути повного фізичного знеструмлення пристрою зі збереженням можливості виконання фонових завдань без ризику розряду АКБ.

Такий підхід дозволяє зберегти простоту базової схемотехніки для поточної реалізації, але водночас забезпечує масштабованість проєкту. Програмна та апаратна підтримка режимів сну з періодичним пробудженням є необхідною умовою для створення універсального автомобільного модуля, що відповідає сучасним вимогам до енергоефективності.

2.2.7 Узагальнення вимог до розроблюваної системи

У попередніх пунктах роботи проведено комплексний аналіз необхідних характеристик: від фізичної реалізації апаратної частини та підтримки специфічних протоколів передачі даних (CAN, K-Line) до забезпечення стабільного живлення та

можливості подальшого нарощування функціоналу системи за рахунок модульної архітектури. Узагальнюючи, можна виділити п’ять базових вимог до системи:

Система повинна відповідати п’яти базовим вимогам:

- має одночасно працювати з двома незалежними CAN-каналами, не перемикаючи їх по черзі;
- повинна підтримувати окремий K-Line-канал для діагностичних процедур;
- користувацький інтерфейс має бути автономним, без обов’язкової прив’язки до смартфона чи ноутбука, зі зручним керуванням за допомогою кнопок або енкодера;
- повинна підтримувати широкий діапазон напруг живлення автомобільної мережі з перешкодами, кидками напруги, ризиком помилкової полярності;
- система має забезпечувати можливість програмного розширювання, тобто додавання нових екранів, груп параметрів і логіки декодування не повинно вимагати заміни апаратної платформи.

За сукупністю вимог можна сформулювати таку інженерну задачу: система має бути компактною, достатньо продуктивною для одночасного обслуговування двох SPI-контролерів CAN і одного UART-сумісного діагностичного каналу, мати простий користувацький інтерфейс та енергоспоживання, яке допускає безперервну роботу від бортової мережі ТЗ без значного нагрівання.

2.3 Розробка структурної схеми системи

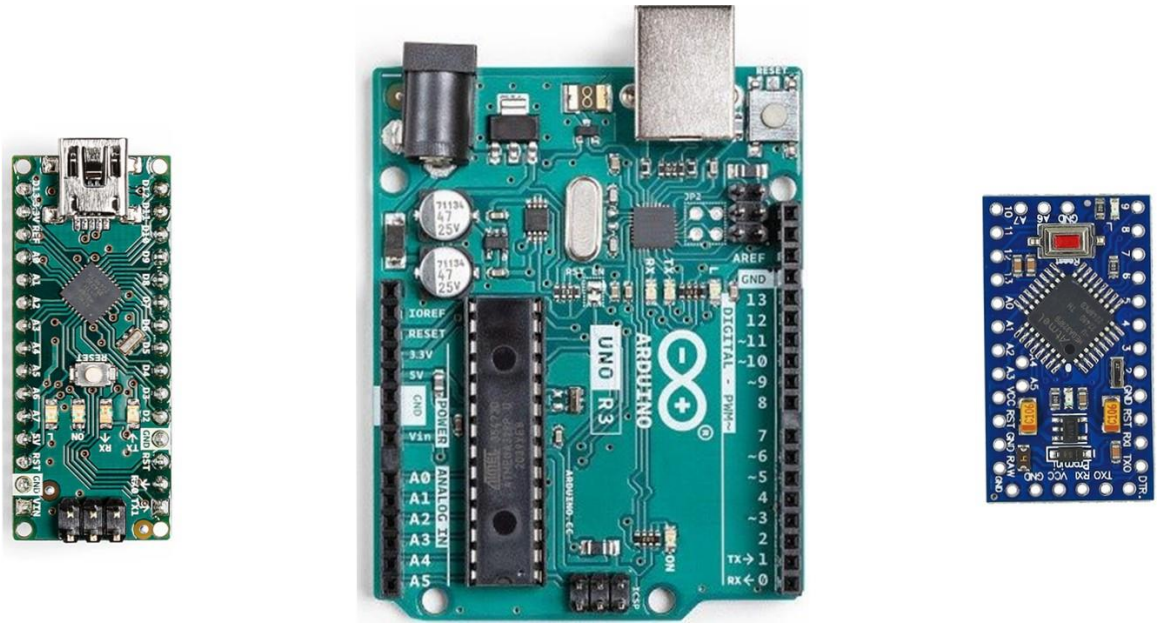
Розглянемо шляхи реалізації і конкретні апаратні та програмні засоби, що забезпечуватимуть задоволення сформульованих вимог. Система базуватиметься на модульній архітектурі, що дозволить забезпечити паралельну обробку даних з різних типів шин та гнучкість при налаштуванні інтерфейсу користувача.

2.3.1 Вибір апаратної платформи

Для реалізації центрального вузла системи доцільно виконати порівняння кількох поширених мікроконтролерних платформ, придатних для обробки даних з двох незалежних CAN-каналів, K-Line інтерфейсу, локального дисплея та вузла вимірювання напруги бортової мережі.

У порівнянні брали участь налагоджувальні плати на базі трьох найбільш поширених апаратних платформ: Arduino, STM32, ESP32. Порівняння виконано за однаковими критеріями: архітектура та тактова частота, обсяг пам’яті, кількість ліній введення-виведення, наявність апаратних інтерфейсів, логічні рівні живлення, а також енергетичні та конструктивні особливості налагоджувальних плат. При цьому для платформи на базі МК ATmega328P слід розрізняти характеристики самого МК та характеристики конкретної плати, оскільки ці плати використовують спільну елементну базу, але відрізняються розмірами, схемою живлення тощо.

Arduino Nano (рис. 2.2, *а*) побудована на МК ATmega328 і працює на частоті 16 МГц. На рівні плати вона має 20 цифрових ліній введення-виведення, 8 аналогових входів, mini-USB інтерфейс та компактний форм-фактор 45×18 мм. Сам МК ATmega328P містить 32 КБ Flash-пам’яті, 2 КБ SRAM та 1 КБ EEPROM, підтримує UART, I^2C і SPI, а також характеризується низьким енергоспоживанням. Важливо зазначити, що в офіційному даташиті Arduino Nano числове значення потужності споживання всієї плати не наведене, тому для коректного порівняння енергоспоживання слід використовувати параметри самого МК, а вплив USB-інтерфейсу, стабілізатора та світлодіодів розглядати окремо [4].



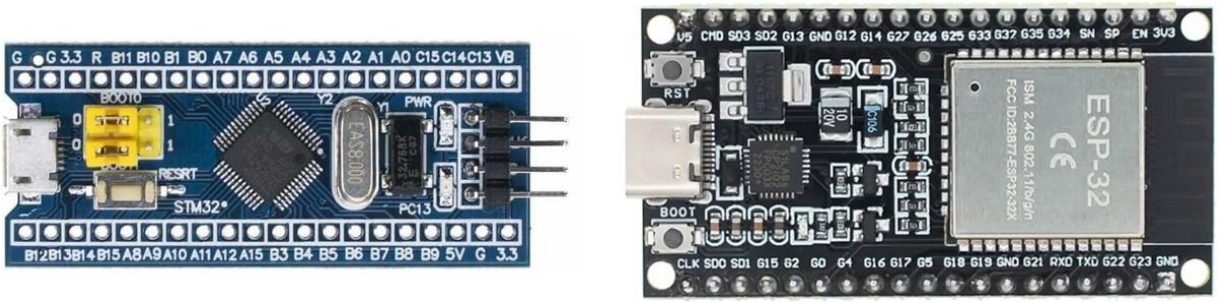
а) *б*) *в*)

Рисунок 2.2 — Зовнішній вигляд налагоджувальнух плат Arduino Nano (*а*), Arduino Uno R3 (*б*) та Arduino Pro Mini (*в*) у виконанні 5 В / 16 МГц.

Arduino Uno R3 (рис. 2.2, *б*) також базується на ATmega328P і, відповідно, має ті самі обчислювальні ресурси: 16 МГц, 32 КБ Flash, 2 КБ SRAM та 1 КБ EEPROM. Плата має 14 цифрових ліній і 6 аналогових входів, USB-B роз'єм та окремий USB-мост на ATmega16U2. Отже, порівняно з Nano, Uno R3 не дає приросту за продуктивністю, а відрізняється переважно більшими габаритами та менш зручним для вбудованого автомобільного модуля форм-фактором. Таким чином, Uno R3 є корисною налагоджувальною платформою загального призначення, але не має суттєвих переваг саме для компактного автономного пристрою в салоні ТЗ [5].

Arduino Pro Mini у виконанні 5 В / 16 МГц (рис. 2.2, *в*) також використовує ATmega328P, тобто з погляду обчислювальних можливостей близька до Nano. Її перевагами є мінімальні габарити 18 × 33 мм, відсутність зайвих вузлів та наявність 14 цифрових і 8 аналогових ліній. Однак Pro Mini не має вбудованого USB-інтерфейсу та потребує зовнішнього FTDI-адаптера для програмування і налагодження. Тому ця плата є привабливою для фінального компактного виробу, але на етапі розробки і лабораторного макетування поступається Nano за зручністю підключення, перепрошивки та сервісного обслуговування [2].

Плата STM32F103C8T6 (рис. 2.3, *а*) представляє вже 32-бітний клас мікроконтролерів і забезпечує істотно більший ресурс: ядро ARM Cortex-M3 із частотою до 72 МГц, 64 КБ Flash, 20 КБ SRAM, близько 37 GPIO на типовій налагоджувальній платі, 10 аналогових входів, 3 USART, 2 I^2C, 2 SPI та один вбудований CAN-контролер. З цієї точки зору STM32F103C8T6 є потужнішою і технічно гнучкішою платформою. Водночас для розроблювальної системи вирішальним є не лише загальний обчислювальний запас, а й відповідність конкретній архітектурі. Оскільки пристрій має одночасно працювати з двома незалежними CAN-сегментами, одного вбудованого CAN-контролера недостатньо, а отже навіть при виборі STM32 все одно довелося б додавати зовнішній CAN-контролер або іншу апаратну надбудову. Крім того, ця платформа є 3,3-вольтовою, що потребує іншої організації живлення всієї системи: роздільного живлення МК та зовнішніх модулів, дисплею тощо [27].



а) *б*)

Рисунок 2.3 — Зовнішній вигляд плат STM32F103C8T6 (*а*) та ESP32-WROOM-32E (*б*).

Представник платформи ESP32 плата ESP32-WROOM-32E (рис. 2.3, *б*) — є ще однією сучасною 32-бітною платформою, що має двоядерний процесор із тактовою частотою до 240 МГц, 520 КБ SRAM, 34 програмовані GPIO, до 18 ADC-каналів, 2 I^2C, 3 UART, 4 SPI, а також вбудовані модулі Wi-Fi та Bluetooth. За сукупністю можливостей це найбільш функціональний варіант серед порівнюваних платформ. Проте саме для даної задачі його переваги не є вирішальними. По-перше, ESP32, як і STM32F103C8T6, має лише один CAN-сумісний TWAI-інтерфейс, тому вимога одночасної роботи з двома CAN-каналами все одно не виконується без зовнішнього апаратного розширення. По-друге, бездротові модулі у цій системі не є обов’язковими, але вони пов’язані з більш високим енергоспоживанням у активних режимах, що є небажаним для простого автомобільного діагностичного модуля [10].

Основні параметри налагоджувальних плат, які приймали участь у порівнянні, наведені в табл. 2.1. За результатами порівняння можна зробити висновок, що Arduino Nano не є формально «найпотужнішою» платформою, однак саме вона забезпечує найкращий баланс між достатністю ресурсів, компактністю, 5-вольтовою логікою, простотою інтеграції та зручністю розробки. Порівняно з Uno R3 вона компактніша за однакових обчислювальних можливостей; порівняно з Pro Mini — зручніша на етапі налагодження завдяки вбудованому USB; порівняно зі STM32 та ESP32 — поступається запасом продуктивності, але не вимагає переходу на іншу логіку живлення і при цьому повністю покриває потреби даної розробки.

Отже, враховуючи переваги і недоліки розглянутих платформ, для даного проєкту обрано платформу Arduino Nano, однак при подальшому розвитку проєкту доцільним може бути використання більш продуктивної платформи ESP32.

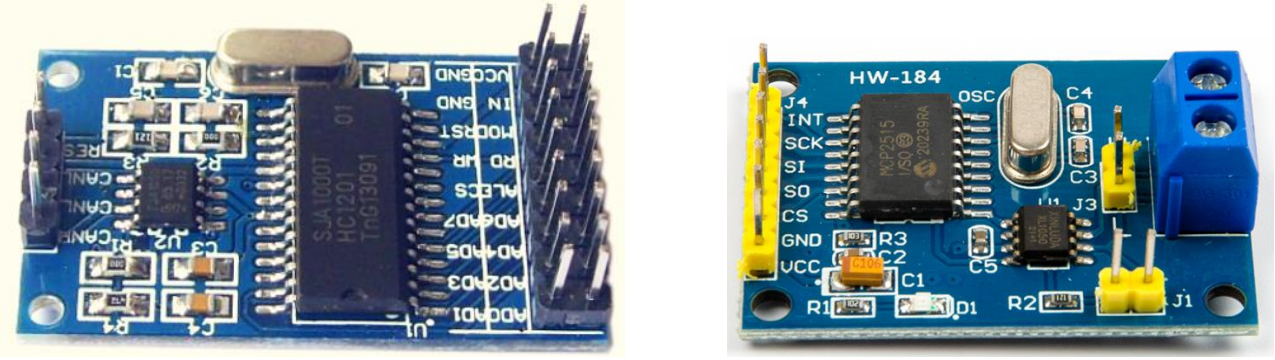
Таблиця 2.1 — Порівняльна характеристика апаратних платформ.

Платформа	Логіка та живлення	Тактова частота та ядро	Пам’ять	Лінії введення-виведен.	Інтер-фейси
Arduino Nano	Плата 5 В, mini-USB, 45×18 мм	ATmega328, 16 МГц, 8-бит AVR	32 КБ Flash, 2 КБ SRAM, 1 КБ EEPROM	20 цифр., 8 анал.	UART, I^2C, SPI
Arduino Uno R3	Плата 5 В, USB-B, ATmega16U2	ATmega328P, 16 МГц	32 КБ Flash, 2 КБ SRAM, 1 КБ EEPROM	14 цифр., 6 анал.	UART, I^2C, SPI
Arduino Pro Mini 5V/16MHz	Плата 5 В, без USB, зовнішн. FTDI; 18×33 мм	ATmega328P, 16 МГц	32 КБ Flash, 2 КБ SRAM, 1 КБ EEPROM	14 цифр., 8 анал., 6 PWM	UART, I^2C, SPI
STM32F103 C8T6 dev. board	Логіка 3,3 В, board doc: 2,7–3,6 В, micro-USB, 23×55мм	ARM Cortex-M3, до 72 МГц	64 КБ Flash, 20 КБ SRAM	37 GPIO, 10 анал. виводів	USART, 2 I^2C, 2 SPI, 1 CAN
ESP32 DevKitC V4 / ESP32-WROOM-32E	Логіка 3,3 В; micro-USB, USB-UART, живлення від 5 В та 3,3 В	Dual-core Xtensa LX6, до 240 МГц	520 КБ SRAM, 448 КБ ROM	34 GPIO, до 18 каналів ADC, 2 DAC	4 SPI, 2 I^2C, UART, Wi-Fi, BT

2.3.2 Вибір контролерів інтерфейсів

Для роботи з шиною CAN було розглянуто декілька технічних рішень. Основними конкурентами на ринку є контролер SJA1000 (NXP), контролери серії MCP2515 від Microchip та інтегровані рішення в сучасних МК.

Класичний контролер SJA1000 (NXP), що використовувався в промисловості достатньо тривалий час (рис. 2.4, *а*), потребує паралельної шини даних (8-біт), що займає майже всі виводи Arduino Nano. Перевагами цього контролера є наявність режиму PeliCAN, розширені функції діагностики, висока надійність, перевірена десятиліттями. Недоліком є потреба в великій кількості виводів МК для паралельної шини адреси/даних, а також високе енергоспоживання порівняно з аналогічними платами. Найчастіше застосовується у промисловому обладнанні [26].



а) *б*)

Рисунок 2.4 — Модулі CAN на основі контролерів SJA1000 (*а*) і MCP2515 (*б*).

Контролер MCP2515 (Microchip) — найпопулярніший дискретний контролер у світі DIY та Arduino (рис. 2.4, *б*). Працює через інтерфейс SPI, що дозволяє підключати його практично до будь-якого сучасного мікроконтролера, що має невелику кількість виводів. Перевагами цього контролера є простота підключення, величезна база готових бібліотек, компактність. До недоліків можна віднести те, що пропускна спроможність обмежена швидкістю SPI та накладними витратами на передачу команд. При щільному потоці даних на шині CAN 1 Мбіт/с контролер може «захлинатися» через малий об’єм апаратного буфера. Найчастіше використовується саме у автомобільних адаптерах, системах моніторингу [23].

Сучасні мікроконтролери (наприклад, STM32 серії F/G/H або ESP32) мають вбудований CAN-контролер безпосередньо на кристалі. Перевагами такого рішення є велика продуктивність за рахунок прямого доступу до пам’яті та роботі на частоті системної шини, підтримка сучасного стандарту Flexible Data-rate (до 64 байт даних у кадрі) та комплектність, оскільки для його роботи потрібен лише зовнішній трансивер (наприклад, TJA1050 або MCP2551). Недоліками є складність ініціалізації (потрібне налаштування тактування шини та таймінгів бітів), а також наявність лише одного подібного блоку у МК. Застосовуються у сучасних блоках керування ТЗ, телематиці, високошвидкісних промислових мережах.

Спроба обійтися одним вбудованим модулем із комутацією на рівні роз’ємів або аналогових ключів призвела б до втрати синхронності, пропусків повідомлень і суттєвого ускладнення ПЗ. Отже, застосування двох зовнішніх контролерів MCP2515 є найбільш раціональним рішенням, оскільки вони повністю

розвантажують центральний процесор від рутинних операцій приймання пакетів, а також забезпечують апаратну фільтрацію ідентифікаторів, що дозволяє ігнорувати непотрібний трафік шини даних та зосередитися на важливих параметрах двигуна. Також вони працюють з логікою 5В, що ідеально узгоджується з обраною платформою Arduino Nano.

2.3.3 Реалізація модуля K-Line

Для реалізації фізичного рівня K-Line було розглянуто три підходи (рис. 2.5): використання спеціалізованого автомобільного трансивера L9637D, застосування мікросхеми MC33199 або побудова дискретного вузла на базі компаратора LM393. Перші два варіанти є готовими інтерфейсними мікросхемами, розробленими саме для роботи з K-Line за ISO 9141, тоді як рішення на LM393 є лише функціональним наближенням, яке потребує додаткових зовнішніх елементів для формування рівнів, захисту та узгодження з бортовою мережею.

а) *б*) *в*)

Рисунок 2.5 — Апаратні засоби реалізації фізичного рівня K-Line:
а – спеціалізований автомобільний трансивер L9637D,
б – мікросхема MC33199 виробництва NXP,
в – подвійний компаратор LM393 виробництва STM.

Мікросхема L9637D (рис. 2.5, *а*) виробництва STMicroelectronics є спеціалізованим ISO 9141 сумісним інтерфейсом і наразі залишається активним продуктом у серійному виробництві. Вона працює в широкому діапазоні напруг 4.5–36 В, витримує автомобільні перехідні процеси до 40 В, має захист від зворотної полярності до -24 В, обмеження струму K-лінії, тепловий захист, контрольовану швидкість фронтів для зниження електромагнітних завад та дуже малий струм у станах off/standby [20], що добре відповідає умовам роботи в бортовій мережі ТЗ.

Мікросхема MC33199 (рис. 2.5, *б*) виробництва NXP також є повноцінним автомобільним ISO 9141 драйвером K- і L-ліній. Його перевагами є підтримка швидкості обміну до 200 кбод, наявність внутрішнього генератора опорної напруги, захист від КЗ та перегріву, робочий температурний діапазон -40…+125 °C [22]. Проте суттєвим недоліком є те, що NXP позначає цей компонент як «archived», тобто він не є рекомендованою позицією для нового серійного застосування.

Рішення на LM393 (рис. 2.5, *в*) є привабливим з точки зору простоти та вартості, однак LM393 за своєю природою є лише подвійним компаратором. Його даташит [21] вказує широкий діапазон живлення, низьке споживання та швидку реакцію, але він не містить інтегрованого драйвера K-Line, не забезпечує готової відповідності ISO 9141 і не має тих захистів, які спеціалізовані мікросхеми. Тому при побудові K-Line на LM393 необхідно окремо реалізовувати формування рівнів, струмовий захист, захист від перенапруг і КЗ.

З огляду на наведене, для розроблюваної системи доцільно обрати саме L9637D, яка має вищу завадостійкість і кращу відповідність умовам автомобільної експлуатації. Але з огляду на доступність LM393, на етапі макетування та лабораторних випробувань є сенс використовувати саме цю мікросхему.

2.3.4 Реалізація користувацького інтерфейсу

У системі моніторингу, що розробляється, ключовим завданням є надання водієві оперативної інформації про стан параметрів ТЗ у режимі реального часу. При виборі типу дисплея необхідно враховувати вимоги щодо інформативності, енергоспоживання та складності програмної підтримки.

На початковому етапі реалізації проєкту основний обсяг даних, що передаються, складається з текстових повідомлень, числових значень параметрів (напруга, температура, ідентифікатори пакетів) та статусних сповіщень. У зв’язку з цим, найбільш доцільним є використання символьного HD44780-сумісного РК-дисплею формату 16x2 (рис. 2.6). Даний тип дисплею забезпечує оптимальне співвідношення між габаритними розмірами пристрою та обсягом інформації, що виводиться. Він характеризується низьким енергоспоживанням та високою контрастністю, що важливо для зчитування даних під час руху автомобіля.

Рисунок 2.6 — Символьний HD44780-сумісний РК-дисплей формату 16x2.

Такий дисплей дозволяє відобразити до 4 параметрів одночасно (по 2 у кожному рядку) при використанні текстових позначень, а при використанні піктограм для позначення температури, обертів, напруги, тощо дозволяє розширити кількість параметрів до 6 одночасно (по 3 у кожному рядку). Це забезпечує водія повною картиною стану ключових вузлів без необхідності перемикання екранів.

Перевагами даного рішення є:

- низьке обчислювальне навантаження: виведення символьної інформації не потребує великих ресурсів оперативної пам’яті МК;
- простота інтеграції: наявність стандартних бібліотек та можливість підключення через інтерфейс I^2C (за допомогою розширювача PCF8574) дозволяють мінімізувати кількість задіяних виводів контролера;
- економічна ефективність: даний тип індикаторів має оптимальне співвідношення ціни та функціональності.

HD44780-сумісні модулі мають відпрацьовані бібліотеки, просту схему ініціалізації та високу сумісність із 5-вольтовими мікроконтролерами. Також вони стабільно працює в температурних межах типових автомобільних застосувань і не вимагає високошвидкісного інтерфейсу чи додаткового відео буфера.

Попри те, що символьного дисплея достатньо для поточних завдань, архітектура системи передбачає можливість модернізації. Як перспективний напрямок розглядається перехід на графічні OLED дисплеї, наприклад, OLED 16×2 (рис. 2.7) з роздільною здатністю 128x64 точки, або більш місткий LCD 20×4, який вміщує більше інформації, але збільшує габарити панелі та відволікає водія.

Рисунок 2.7 — Графічний OLED-дисплей SSD1306 I^2C 0.91" 128x32.

Можливості розвитку при використанні графічного дисплея наступні: побудова графіків зміни параметрів у часі, що спрощує діагностику перехідних процесів; використання шрифтів різного розміру, іконок стану та створення багаторівневих графічних меню; відображення більшого обсягу даних на аналогічній фізичній площі екрана.

Таким чином, для базової версії системи обирається символьний HD44780-сумісний дисплей 16x2, як такий, що повністю відповідає поточному технічному завданню. При цьому програмно-апаратна частина проєктується з урахуванням можливості подальшої адаптації під графічні інтерфейси виведення даних без докорінної переробки структури пристрою.

Автоматичне регулювання яскравості дисплея доцільно реалізувати за сигналом від датчика освітленості, підключеного до аналогового входу мікроконтролера, з подальшим керуванням підсвічуванням за допомогою ШІМ і ключового транзистора, або спеціалізованої плати керування. Такий підхід дозволяє плавно змінювати яскравість у денному та нічному режимах, підвищує читабельність екрана і зменшує засліплення водія в темний час доби.

Для оперативного інформування водія про перевищення порогів і критичні стани в системі доцільно передбачити звуковий випромінювач (буззер). Якщо буде використовуватись активний буззер, його можна вмикати цифровим сигналом через транзисторний ключ; якщо пасивний — доцільно використати керування за допомогою ШІМ для формування різних тонів і шаблонів попередження. Звукова індикація повинна використовуватися лише для пріоритетних подій, щоб не перевантажувати водія зайвими сигналами.

Для введення команд і зміни налаштувань можуть використовуватися кнопки, енкодер або сенсорна панель. На етапі макетування найбільш доцільним є

використання кнопок або енкодера, оскільки такі рішення простіші в інтеграції, залишаються зручними в умовах вібрацій і не вимагають графічного дисплея. Пристрої введення повинні забезпечувати циклічну навігацію по сторінках, зміну порогових значень. Використання сенсорної панелі є доцільним лише у випадку переходу на графічний інтерфейс із великою площею екрана.

2.3.5 Реалізація періодичного контролю

Для реалізації часових міток, періодичного пробудження системи, режимів сну та ведення журналу подій доцільно застосувати годинник реального часу (RTC). Однак, обрана апаратна платформа ArduinoNano, на відміну від плат платформи ESP32 або STM32, не включає вбудований RTC. У цьому випадку треба використовувати зовнішній модуль RTC на основі DS1307 або DS3231 (рис. 2.8). Мікросхема DS1307 є простим RTC з інтерфейсом I²C, резервним живленням та 56 байтами SRAM, однак працює із зовнішнім кварцовим резонатором і поступається за точністю часової бази. Мікросхема DS3231, навпаки, містить інтегровані термокомпенсований кварцовий генератор і резонатор, має сигнали alarm/reset і забезпечує значно вищу точність у робочому температурному діапазоні.

а) *б*)

Рисунок 2.8 — Модулі годинника реального часу на DS1307 (*а*) та DS3231 (*б*).

З огляду на необхідність стабільного відліку часу в умовах змін температури всередині автомобіля, для розроблюваної системи доцільно обрати саме модуль на DS1307. Такий вибір спрощує апаратну реалізацію, підвищує точність часових міток і дозволяє реалізувати апаратне пробудження МК за сигналом RTC.

2.3.6 Система живлення

Усі основні вузли розроблюваної системи працюють від напруги 5 В. Це дозволяє побудувати систему з єдиною шиною живлення і уникнути окремих рівневих доменів. З іншого боку це висуває жорстку вимогу до якості живлення: автомобільна мережа має перешкоди, комутаційні імпульси, провали під час запуску двигуна та ризик переполюсування.

Для забезпечення стабільної роботи системи може бути використано імпульсний DC-DC перетворювач LM 2596 (рис. 2.9, *а*) [28]. Перевага цього перетворювача полягає у великому запасі по струму до 3 А, а також у значно нижчих теплових втратах порівняно з лінійними стабілізаторами типу 7805. Вхідний діапазон до 40 В і проста типова схема підключення роблять LM 2596 найбільш зручним варіантом.

а) *б*)

Рисунок 2.9 — Варіанти організації живлення СК:
а — імпульсний DC-DC понижуючий перетворювач на базі LM2596,
б — гальванічно ізольований промисловий модуль URB2405YMD.

Лінійні стабілізатори LM 2940 або 7805 можуть розглядатися як альтернативи, але лише в обмежених сценаріях: LM 2940 є значно кращим за 7805 для автомобільного застосування, має захист від зворотної полярності та придатний як післястабілізатор, однак для живлення всієї системи він програє перетворювачу LM 2596 за розсіюваною потужністю.

Альтернативою розглянутому вище DC-DC перетворювачу на LM2596 може виступати гальванічно ізольований промисловий модуль Mornsun URB2405YMD (рис. 2.9, *б*), розроблений для складних умов експлуатації, перевагами якого є наявність гальванічної розв’язки (вхідні та вихідні кола розділені трансформатором,

що критично важливо в автомобілях або промисловості для захисту логічної частини від стрибків напруги в силовій мережі), надійність (має повний комплекс вбудованих захистів, якщо на вході виникне замикання, модуль просто вимкнеться, зберігши працездатність), стабільність (широкий діапазон входу (9–36В) дозволяє системі стабільно працювати навіть при значних просіданнях напруги під час пуску двигуна) [29]. Недоліками цього модуля є значно вища ціна (у 5-10 разів дорожче за модулі LM2596).

Отже, для серійного випуску пристрою моніторингу вибір модуля URB2405YMD-10WR3 є більш доцільним оскільки він має гальванічну ізоляцію (гарантує, що дороговартісні компоненти не вийдуть з ладу при стрибках напруги у мережі живлення), відповідає стандартам та є компактним (модуль монтується безпосередньо на друковану плату, що робить кінцевий пристрій монолітним та вібростійким). Однак, на етапі макетування та лабораторних випробувань є сенс використовувати імпульсний DC-DC понижуючий перетворювач на базі LM2596, що має суттєво нижчу вартість.

Для забезпечення стабільної роботи в умовах бортової мережі, система живлення також повинна включати багатоступеневий вузол захисту. Він складається з діода Шотткі (захист від переполюсовки), TVS-діода (для придушення високовольтних імпульсів) та LC-фільтра, який мінімізує вплив електромагнітних завад від системи запалювання. Використання імпульсного перетворювача з такою схемою обв’язки гарантує високий ККД та термічну стабільність пристрою.

2.3.7 Структурна схема системи

Враховуючи сформульовані вимоги до системи контролю і обрані апаратні засоби для її реалізації розроблено структурну схему системи (рис. 2.10), що відображає логічний взаємозв’язок між апаратними модулями пристрою.

Живлення від бортової мережі ТЗ проходить через вхідний захист і надходить на DC/DC перетворювач, який формує стабільну напругу 5 В. Ця напруга подається безпосередньо на загальну шину живлення всіх компонентів. Окремо від бортової мережі напруга подається на резистивний подільовач, який узгоджує напругу бортової мережі з допустимим діапазоном вхідних напруг АЦП МК.

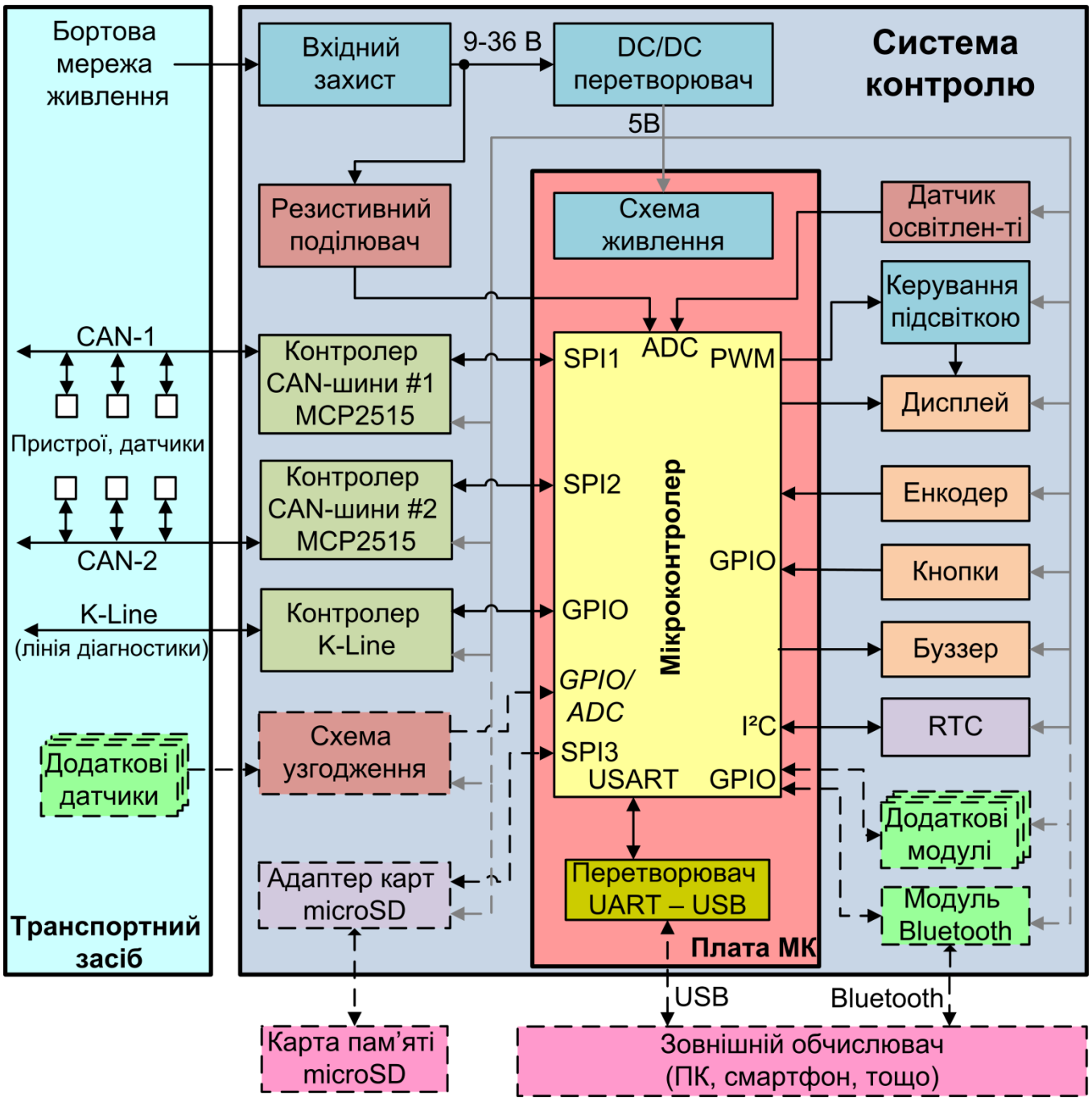


Рисунок 2.10 — Структурна схема системи контролю робочих параметрів ТЗ.

Центром системи є налагоджувальна плата, яка через шину SPI взаємодіє з двома контролерами MCP2515 для моніторингу CAN-шин (кожен контролер має окремі сигнали ChipSelect та Interrupt). Модулі MCP2515 фізично підключаються до двох CAN-шин ТЗ. Окремий канал зв'язку через контролер K-Line, реалізований на основі аналогового компаратора LM393, забезпечує роботу з лінією діагностики.

РК-дисплей працює як локальний інтерфейс користувача, та підключений до мікроконтролера через паралельний інтерфейс у 4-бітному режимі задля економії

портів МК. Керування підсвічуванням дисплея виконується за допомогою окремого блока, яким керує МК на підставі даних з датчика освітлення.

Для керування роботою пристрою та введення необхідних параметрів в системі передбачено використання з кнопками або енкодера з функцією натискання. Живлення ці модулі отримують від загальної мережі живлення, а опитування їх стану виконується МК. Для інформування водія про надзвичайні події передбачено використання буззера, який вмикається також по команді від МК.

Для розширення функціоналу пристрою передбачене підключення модуля реального часу (для можливості реалізації режим сну та/або виведення на екран поточної дати та часу), адаптера карт пам’яті (для реалізації можливості реєстрації пристроєм параметрів протягом часу). Також передбачена можливість підключення інших додаткових модулів, які можуть знадобиться у подальшому.

Підключення зовнішніх датчиків до МК повинне виконуватись через окремі резистивні поділювачі до цифрових або аналогових каналів введення/виведення для приведення рівня сигналу, що вимірюється, до меж, які зможе виміряти МК.

Зв’язок МК із зовнішнім пристроєм (ПК, смартфон) виконується за допомогою вбудованого в плату МК перетворювача UART-USB та стандартного кабелю USB або через інтерфейс Bluetooth за умови використання відповідного модуля.

2.3.8 Структура і функції програмного забезпечення

Програмне забезпечення розробляється за модульним принципом з використанням середовища Arduino IDE. Основні функції ПЗ включають:

- ініціалізація — налаштування швидкостей обміну для обох модулів CAN та встановлення параметрів зв’язку для K-Line; ініціалізація РК-дисплею;
- фонове «прослуховування» шини та фільтрація пакетів. Отримані дані (оберти, температура, навантаження) зберігаються у відповідних змінних;
- виконання діагностичних запитів для отримання специфічних даних (наприклад, пропуски запалювання), які не транслюються в шину CAN постійно, та похибок в режимі зчитування та видалення похибок;
- опитування стану кнопок керування, енкодера та зміна активної «сторінки;
- вивід сформованого рядка показників з використанням піктограм.

На рівні логіки відображення доцільно передбачити декілька сторінок, які можуть перемикатися циклічно: основні параметри двигуна, бортмережі, лічильники, сторінка роботи з похибками.

Прийом CAN-повідомлень має виконуватись через INT-лінії MCP2515. Оновлення дисплея достатньо виконувати в межах 2 разів на секунду. K-Line-опитування не повинно виконуватися безперервно в штатному режимі — лише за запитом. Такий розподіл зменшує ризик пропуску кадрів, усуває «підвисання» інтерфейсу та дозволяє зберегти просту однопотокову архітектуру.

Розроблена блок-схема алгоритму наведена на рис. 2.11. На початку роботи програми виконується ініціалізація, необхідних фільтрів та налаштувань зовнішніх модулів програми; ініціалізація дисплею з формуванням іконок для можливості відображення параметрів. Також виконується ініціалізація апаратних виводів МК на роботу у режимі переривань для можливості інформування МК про надходження нових даних від CAN-шини без періодичного опитування. Головний цикл програми складається з виконання декількох окремих функцій, реалізація яких буде виконуватись за модульним принципом для можливості розширення системи:

- виконується перевірка наявності переривання від одного або двох контролерів CAN-шини. Якщо переривання наявне, то виконується запит до контролерів, зчитування даних, проводиться аналіз даних, виокремлення необхідних чисельних показників, які відображають необхідні параметри роботи двигуна ТЗ, та запис розрахованих даних у змінні програми;
- на наступному етапі виконується опитування кнопок та енкодера. Якщо було натискання однієї з кнопок або прокручування енкодера, то встановлюється потрібний режим роботи програми (режим відображення основних параметрів, режим зчитування помилок, тощо);
- якщо програма знаходиться у режимі зчитування помилок, то виконується запит по K-Line списку зареєстрованих ЕБК помилок;
- далі виконується перевірка, що отримано відклик від ЕБК, зчитування переліку помилок та завантаження отриманих даних у внутрішні змінні;

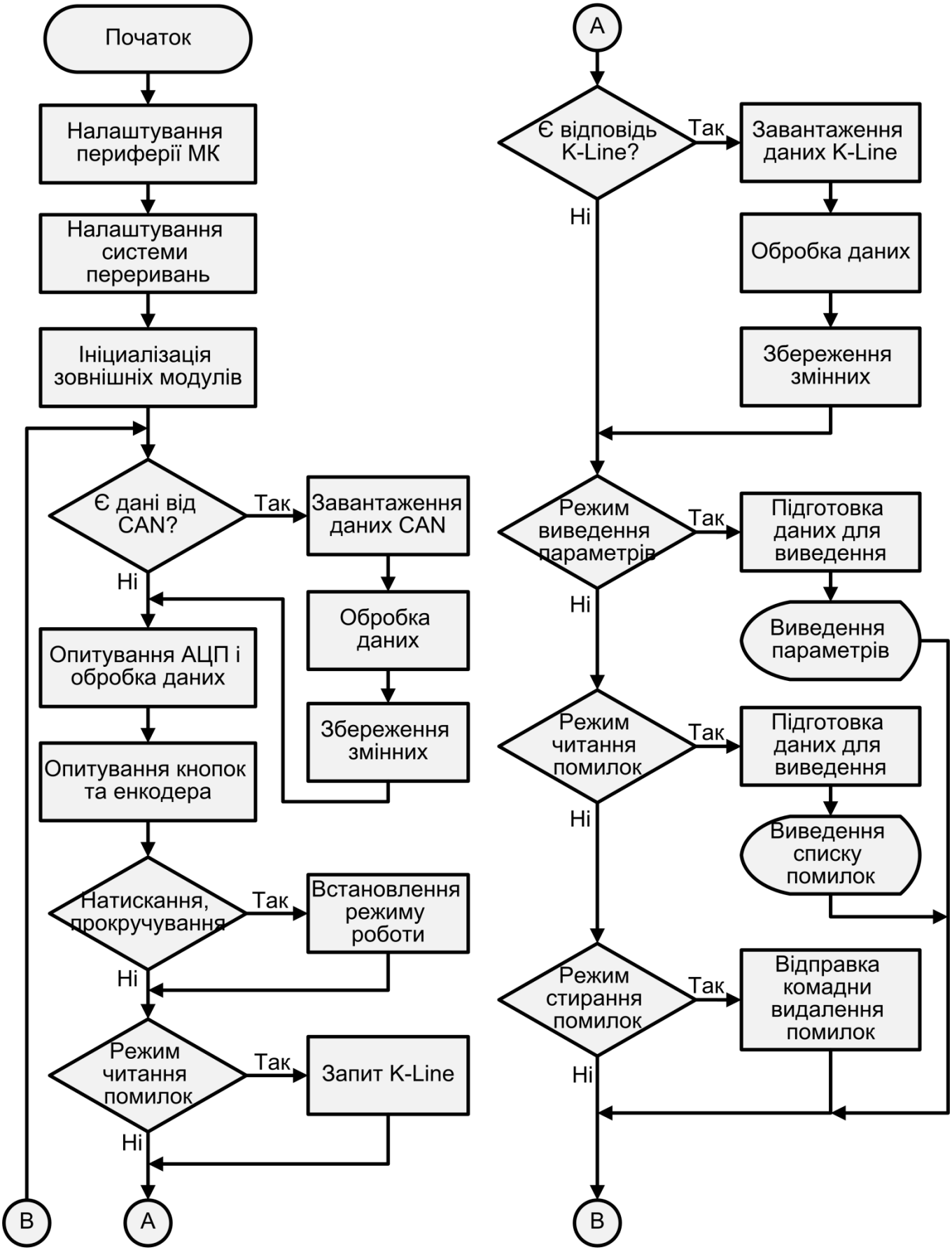


Рисунок 2.11 — Блок-схема програмного забезпечення мікроконтролера.

- на наступному етапі виконується формування текстових рядків даних для виведення на дисплей (у залежності від режиму роботи та активної «сторінки») та виведення даних на дисплей;
- якщо у якості режиму роботи програми обрано режим видалення помилок, то по K-Line відправляється запит до ЕБК з відповідною командою;
- далі виконується перехід до початку циклу та основний цикл програми виконується заново.

Також, для використання пристрою в якості модуля пасивного моніторингу даних для аналізу смислового наповнення вхідних пакетів даних CAN-шини, слід передбачити можливість роботи в режимі відправки вхідних даних на ПК для подальшого аналізу.

2.4 Висновки до розділу

У другому розділі було проведено повний цикл проєктування системи контролю параметрів ТЗ. На основі аналізу особливостей експлуатації автомобілів з комбінованими протоколами обміну (CAN та K-Line) було:

- визначено перелік критичних параметрів для моніторингу, що включає як загальні показники (оберти, температура), так і специфічні діагностичні дані (навантаження, пропуски запалювання);
- сформульовано технічні вимоги до апаратної частини, зокрема щодо вібростійкості, термостабільності та якості живлення в умовах бортової мережі;
- обґрунтовано вибір апаратної платформи та спеціалізованих контролерів, що забезпечують оптимальний баланс між вартістю та функціональністю;
- розроблено структурну схему пристрою, яка передбачає модульну архітектуру з можливістю розширення функціоналу, включаючи режим пасивного моніторингу даних для аналізу на ПК;
- розроблено блок-схему програмного забезпечення МК.

Спроєктована архітектура є фундаментом для подальшої розробки принципових схем та програмного забезпечення.

РОЗДІЛ 3 СТВОРЕННЯ МОДЕЛІ І ДОСЛІДЖЕННЯ РОБОТИ СИСТЕМИ

У даному розділі розглянуто етапи проєктування, програмно-апаратної реалізації, комп'ютерного моделювання та натурного тестування розроблюваної системи. Описано схемотехніку модулів збору даних з бортових шин, обґрунтовано архітектуру ПЗ та алгоритми взаємодії периферійних модулів. Особливу увагу приділено створенню цифрового двійника системи у віртуальному середовищі Wokwi та верифікації розроблених алгоритмів у реальних сценаріях експлуатації ТЗ.

3.1 Розробка апаратної частини системи

Розробка апаратної частини системи базується на принципах модульності, енергоефективності та стійкості до завад бортової мережі ТЗ. Основним завданням апаратного комплексу є надійне детерміноване перехоплення даних із шини CAN та діагностичної лінії K-Line, їх первинна фільтрація, обробка обчислювальним ядром та виведення інформації для водія. До складу розроблюваного пристрою входять центральний мікроконтролерний модуль, два модулі шини CAN, фізичний інтерфейс K-Line, рідкокристалічний модуль індикації, інші додаткові модулі.

3.1.1 Модулі шини CAN

Для підключення до автомобільної шини CAN застосовано спеціалізовані модулі на базі автономного апаратного контролера MCP2515 та трансивера фізичного рівня TJA1050. Контролер MCP2515 повністю реалізує специфікацію CAN версії 2.0B, підтримує як стандартні (11-бітні), так і розширені (29-бітні) ідентифікатори кадрів, а також містить вбудовані апаратні маски та фільтри для сепарації корисних повідомлень без залучення ресурсів центрального процесора.

У відповідності до типової схеми підключення модуля (рис. 3.1), зв'язок між МК та контролером MCP2515 (U2) здійснюється через роз'єм J4 за допомогою високошвидкісного послідовного інтерфейсу SPI (лінії MOSI, MISO, SCK, CS), що працює в режимі Slave для модуля CAN [23]. Додатково використовується лінія переривання (INT) модуля, що має бути підключена до виводу зовнішнього переривання МК. Це забезпечує можливість асинхронного зчитування нових кадрів даних без втрати пакетів у моменти високої зайнятості шини.

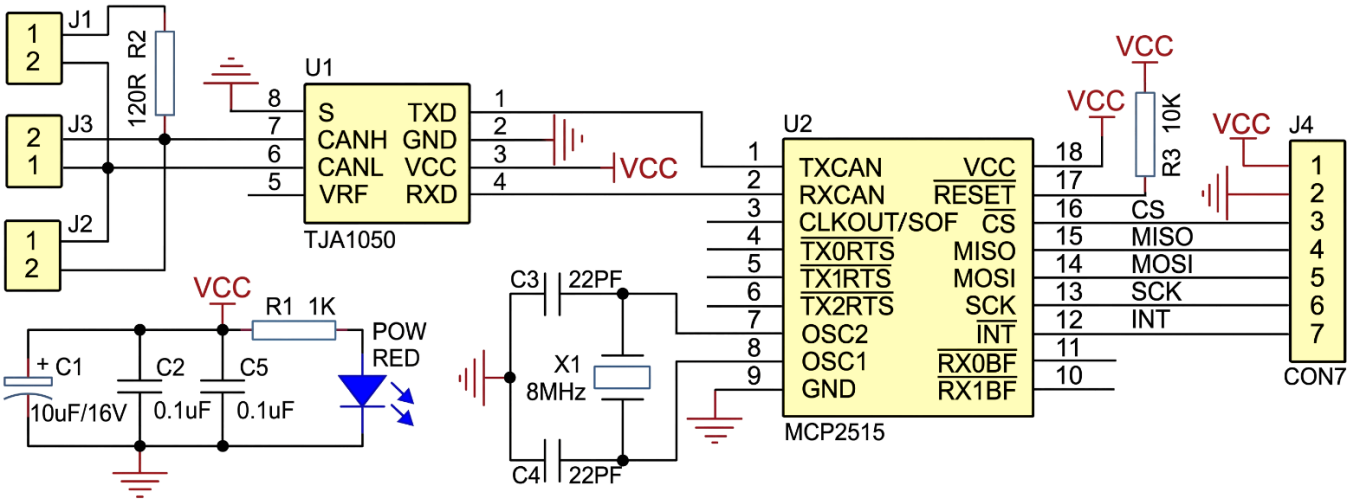


Рисунок 3.1 — Схема підключення контролера MCP2515.

Особливістю архітектури розроблюваної системи є використання двох модулів MCP2515, підключених паралельно до єдиної шини SPI мікроконтролера. Таке рішення зумовлене необхідністю одночасного моніторингу двох ізольованих контурів автомобільної мережі (шин силового агрегату Powertrain CAN зі швидкістю 500 кбіт/с та комфорту Comfort CAN зі швидкістю 100 кбіт/с).

Паралельне підключення можна реалізувати шляхом об’єднання ліній обміну даними MOSI, MISO та тактування SCK. Розділення звернень мікроконтролера до конкретного модуля здійснюється через індивідуальні лінії вибору мікросхеми Chip Select (CS). Для миттєвого зчитування даних без опитування у циклі, виходи переривань (INT) обох модулів мають бути підключені до виводів мікроконтролера, що підтримують зовнішні переривання.

Для фізичного сполучення з диференційними лініями шини автомобіля (CAN-High та CAN-Low) до складу модулів входять мікросхема трансивера TJA1050 (U1). Вони виконують роль інтерфейсу між логічними рівнями контролера MCP2515 (U2) та диференційними рівнями напруги фізичного рівня CAN. Трансивер забезпечує високу завадостійкість, захист від короткого замикання ліній на масу або джерело живлення, а також електромагнітну сумісність у жорстких умовах експлуатації бортової мережі ТЗ.

3.1.2 Модуль зв’язку з шиною K-Line

Оскільки досліджувані транспортні засоби використовують не лише сучасну шину CAN, але й діагностичний протокол K-Line (ISO 9141 / ISO 14230), до складу пристрою має бути інтегровано контролер, працюючий за протоколом UART.

Шина K-Line є однопровідною двоспрямованою лінією із підтяжкою до напруги бортової мережі ТЗ. Модуль на базі LM393, створений за наведеною принциповою схемою (рис. 3.2), виконує функції контурів приймання (Rx) та передачі (Tx). Один канал компаратора LM393 порівнює вхідну напругу з лінії K-Line із опорною напругою, сформованою резистивним дільником та формує чіткий прямокутний TTL-сигнал (0÷5 В), який подається на апаратний вхід Rx МК. Другий канал компаратора трансформує вихідний TTL-сигнал з виводу Tx МК у високовольтні імпульси open-collector, здатні комутувати автомобільну К-лінію.

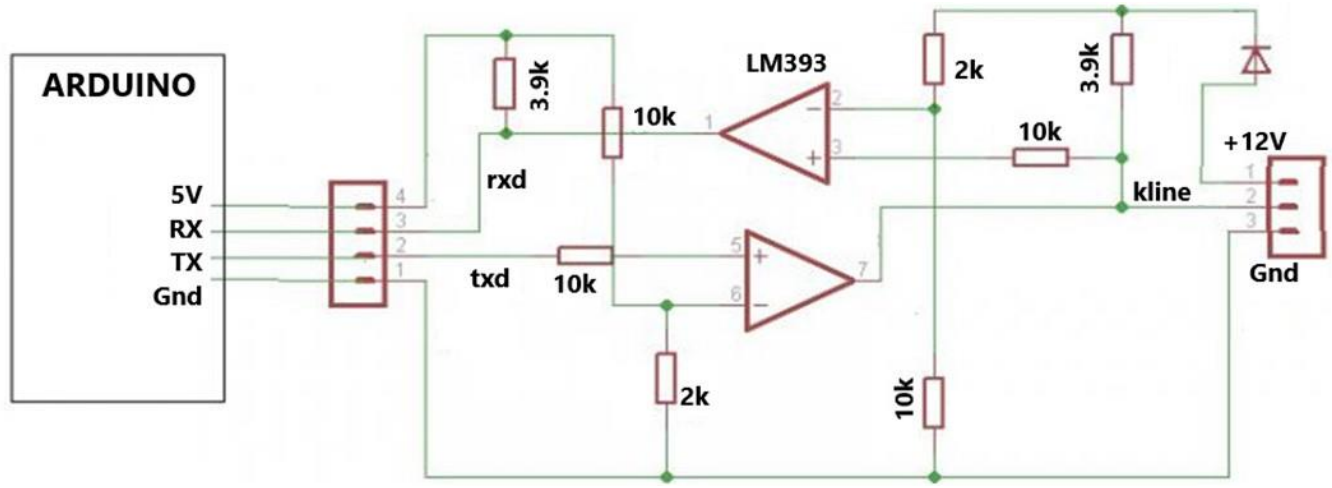


Рисунок 3.2 — Принципова схема модулю зв’язку з шиною K-Line.

3.1.3 Знакосинтезуючий дисплей

Для відображення розрахованих параметрів водію було обрано класичний рідкокристалічний знакосинтезуючий дисплей на базі контролера HD44780 формату 16x2 (два рядки по 16 символів), принципова схема підключення якого до МК наведена на рис. 3.3. З метою економії обмежених виводів МК, підключення дисплея реалізовано за 4-бітною сигнальною схемою. При такій організації дані передаються частинами (ніблами) по 4 біти за один такт стробування, що потребує лише 4 лінії даних (D4–D7) та 2 лінії керування (RS та E). Таким чином, загальна

кількість задіяних портів мікроконтролера становить 6, що залишає достатньо ресурсів для шин CAN та K-Line. Для керування яскравістю дисплею використовується один з виходів МК з можливістю генерації ШІМ-сигналу, корельованого зі значеннями вимірюваного рівня освітлення.

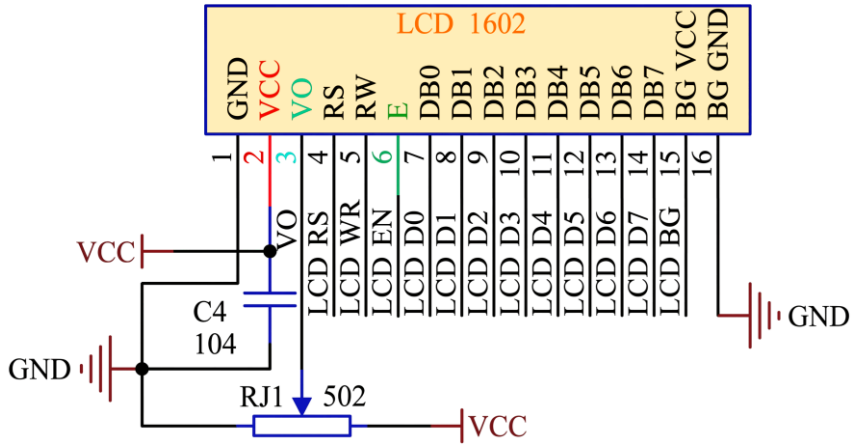


Рисунок 3.3 — Принципова схема підключення дисплею на базі HD44780 до МК.

Незважаючи на використання паралельного 4-бітного інтерфейсу у якості основного, архітектура системи передбачає можливість модернізації та підключення дисплея через інтерфейс I^2C за допомогою дочірньої плати-розширювача портів на базі мікросхеми PCF8574. Це дозволить у перспективі вивільнити 4 цифрові порти мікроконтролера, перевівши керування екраном лише на дві лінії (SDA, SCL), що потребуватиме лише точкової зміни конфігурації програмного коду.

3.1.4 Додаткові апаратні засоби

Окрім наведених в попередніх підрозділах основних модулів, для функціонування контролера необхідні додаткові компоненти, наприклад, резистивний поділювач, датчик освітлення, годинник реального часу, кнопки тощо.

Оскільки аналогові входи мікроконтролера здатні вимірювати напругу лише в діапазоні від 0 до 5 В, для прямого моніторингу вищої напруги бортової мережі слід застосувати схему знижувального резистивного поділювача напруги. Схема складається з двох послідовно з’єднаних прецизійних резисторів R1 (підключається до бортової мережі ТЗ) та R2 (підключається до маси GND). Сигнал, що вимірюється, знімається із середньої точки між резисторами та подається на вивід МК, налаштований на роботу у якості аналогового входу. При цьому для безпечного

вимірювання напруги бортової мережі використовується співвідношення резисторів, що забезпечує коефіцієнт ділення 1:6.

Для визначення рівня зовнішньої освітленості застосовують фоторезистор (напівпровідниковий компонент, який змінює свій омічний опір залежно від інтенсивності падіння світла). У темряві його опір є максимальним (сотні кОм), при яскравому освітленні падає до сотень Ом. Для перетворення зміни опору в зміну напруги, яку може злічити МК, фоторезистор вмикається у схему дільника напруги послідовно з постійним резистором номіналом 10 кОм (рис. 3.4). Один вивід фоторезистора (R2) підключається до шини живлення, другий — до аналогового піна МК, постійний резистор (R1) вмикається між аналоговим піном та заземленням.

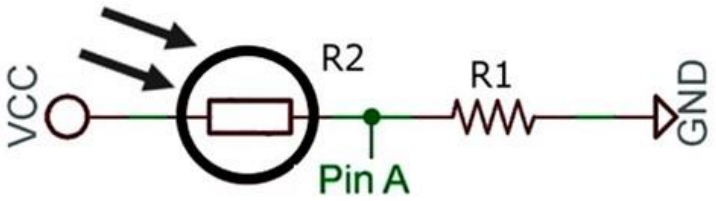


Рисунок 3.4 — Схема підключення фоторезистора до МК.

Для забезпечення незалежного та точного відліку часу в системі, а також можливості пробудження за розладом, можна використати модуль годинника реального часу (RTC) на базі мікросхеми DS1307 (рис. 3.5). Модуль оснащений власним кварцовим генератором з термокомпенсацією та автономним джерелом живлення, що дозволяє зберігати хід часу при відключенні основного живлення системи. Взаємодія такого модуля з МК здійснюється через апаратний послідовний інтерфейс I^2C, який використовує дві лінії зв’язку: послідовну лінію даних (SDA) та послідовну лінію тактування (SCL).

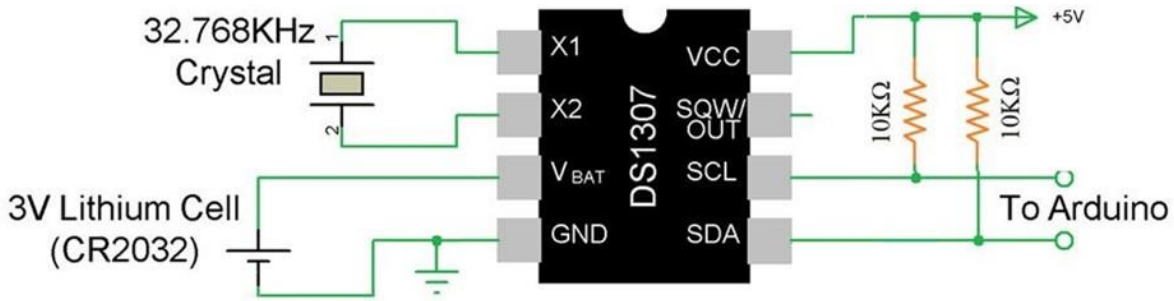


Рисунок 3.5 — Схема підключення RTC на базі мікросхеми DS1307.

Для генерації звукових сповіщень та аварійних сигналів у системі можна застосувати пасивний п’єзоелектричний випромінювач (бузер), який не має генератора та вимагає подачі ШІМ-сигналу (через функцію tone()) для створення звуків різної тональності. Оскільки споживаний струм бузера вкладається в межі допустимого струму на один пін МК (до 20 мА), він підключається напряму: позитивний вивід до піна МК із підтримкою ШІМ, негативний до шини GND.

Для реалізації керування режимами роботи системи застосовується механічна тактова кнопка без фіксації, один вивід якої підключається до цифрового піна МК, другий — до шини GND. При цьому важливо уникати «стану невизначеності» (коли пін «плаває» в повітрі та ловить наводки при розімкненій кнопці), для уникнення якого застосовується схема із вбудованим програмним підтягувальним резистором (у розімкненому стані на піні завжди присутня логічна «1», а при натисканні кнопки пін замикається на землю, викликаючи логічний «0»).

3.1.5 Розподіл периферійних компонентів по виводах МК

Розподіл периферійних компонентів по виводах мікроконтролера ATmega328P (рис. 3.6) не є довільним і зумовлений архітектурними особливостями та апаратними обмеженнями обраної платформи.

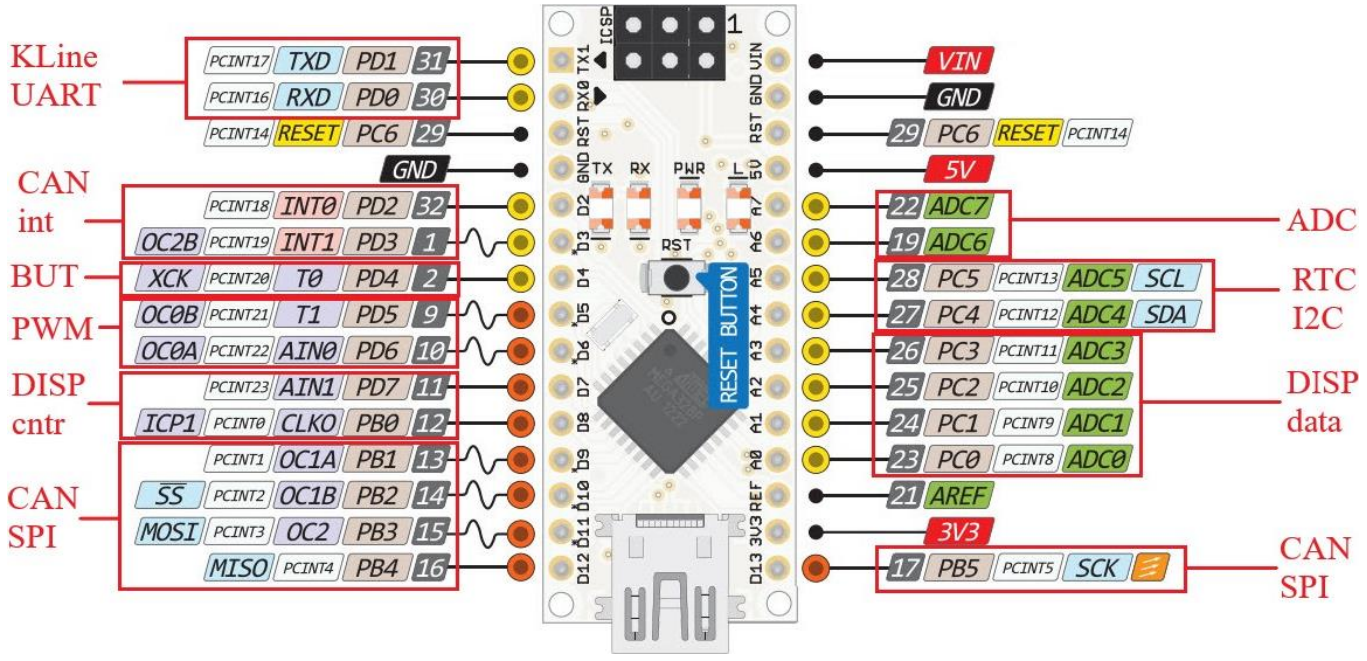


Рисунок 3.6 — Розподіл периферійних компонентів по виводах МК ATmega328P.

Для забезпечення коректної взаємодії всіх модулів було враховано спеціалізовані функції окремих груп виводів. Першочерговим етапом проєктування було резервування пінів, що мають фіксовані апаратні блоки для послідовної передачі даних, оскільки їхня програмна емуляція суттєво завантажує процесор.

Апаратний UART (піни D0 та D1) зазвичай застосовуються для зв'язку МК з ПК (через мікросхему CH340G), і під час роботи комплексу в режимі «логера» апаратний UART відключається та контролер K-Line не використовується. Однак, під час роботи комплекту у режимі контролера немає необхідності у зв'язку з ПК і ці піни застосовуються для зв'язку з контролером K-Line через апаратний UART.

Оскільки модуль годинника реального часу (RTC DS1307) працює за апаратним протоколом I^2C/TWI, який в обраному МК реалізований виключно на пінах A4 та A5, цей модуль має бути підключений саме до цих виводів, що дозволить використовувати апаратний блок МК для швидкого обміну даними.

Апаратна реалізація протоколу обміну SPI (піни D11, D12, D13) зарезервована для підключення високошвидкісної периферії, яка працює за цим протоколом — модулів зв'язку з CAN-шиною. Також ці модулі потребують окремого підключення ліній CS кожного модуля (обрано цифрові піни D9 та D10) та окремого підключення ліній INT (використовуються піни з апаратними перериваннями D2 та D3).

Виводи АЦП A6 та A7 можуть використовуватись лише у якості аналогових входів, отже саме ці виводи обрано для підключення компонентів, що видають аналоговий сигнал та потребують оцифрування рівнів напруги: резистивного дільника напруги (пін A6) та датчика освітленості (пін A7).

Окрім лінії регулювання яскравості, для підключення РК дисплею необхідно використовувати 6 цифрових пінів (4 для лінії даних та 2 для лінії керування). Отже, для лінії даних використано піни A0, A1, A2, A3 (налаштовані на роботу в режимі цифрових виводів); для лінії керування — піни D7 та D8.

Цифровий пін D4 буде використовуватись для підключення кнопки керування. Виводи D5 та D6, які підтримують апаратну ШІМ обрано для підключення п'єзовипромінювача та вивода регулювання яскравості РК-дисплею відповідно.

3.2 Розробка програмного забезпечення системи

Програмний комплекс системи розроблено на засадах модульності та ієрархічної структуризації. Основний фокус при проєктуванні ПЗ було зроблено на забезпеченні детермінованості обробки вхідних переривань від двох модулів CAN та чіткого таймінгу ініціалізації сесій зв'язку K-Line. Програма працює в циклічному режимі з розподілом пріоритетів: найвищий пріоритет мають підпрограми обробки апаратних переривань шин обміну, середній — розрахункові алгоритми, нижчий — оновлення текстових полів екранного інтерфейсу.

3.2.1 Середовище розробки Arduino IDE

Розробка, компіляція та відлагодження програмного коду здійснювалися у кросплатформному відкритому інтегрованому середовищі розробки Arduino IDE (рис. 3.7). Програмний код написаний мовою C++ із використанням низькорівневих маніпуляцій з регістрами МК для оптимізації швидкодії операцій вводу-виводу.

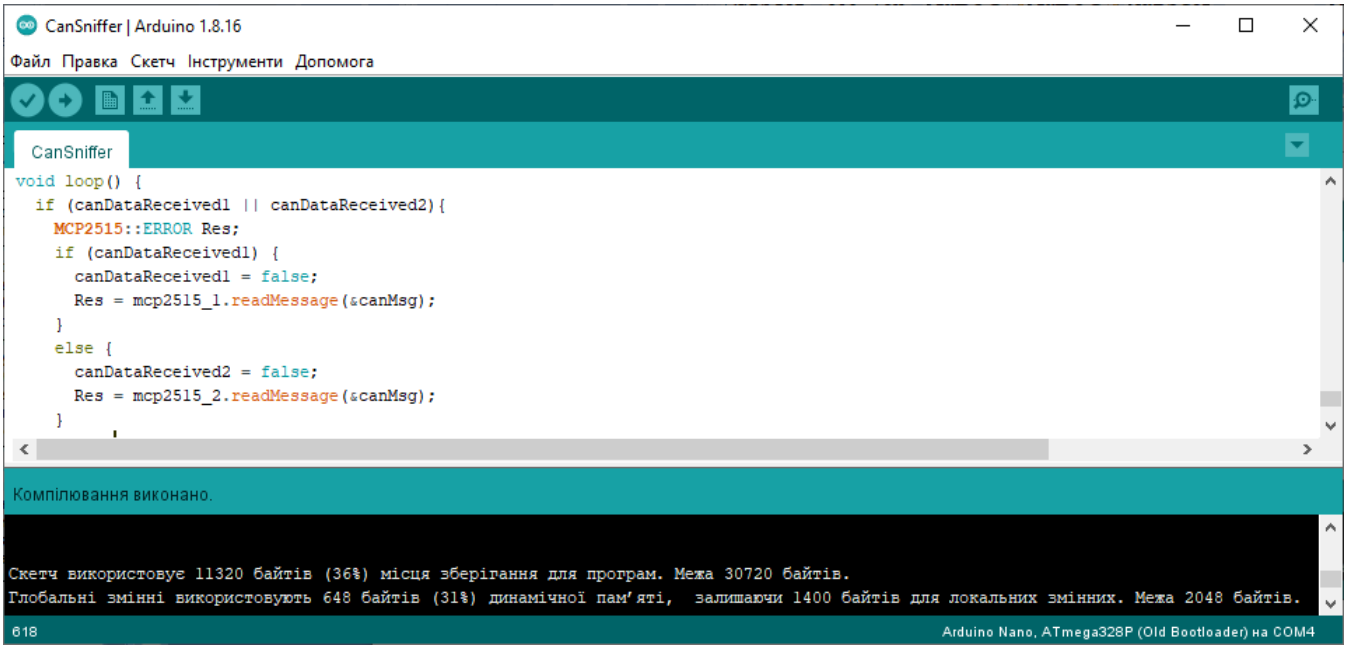


Рисунок 3.7 — Відкрите інтегроване середовище розробки Arduino IDE.

Програмне середовище Arduino IDE забезпечило зручний інструментарій для підключення зовнішніх файлів, керування завантаженням прошивки через вбудований USB-UART міст плати Arduino Nano, а також надало доступ до монітора послідовного порту (Serial Monitor). Останній активно використовувався на етапах первинного відлагодження для виведення сирих HEX-дампів автомобільних шин.

3.2.2 Бібліотеки, що використовуються в проєкті

Програмне забезпечення нижнього рівня для взаємодії з апаратним контролером CAN базується на використанні оптимізованої бібліотеки «mcp_can». Ця бібліотека надає набір високорівневих абстракцій для конфігурування регістрів мікросхеми MCP2515, ініціалізації режимів роботи та безпосереднього приймання й передачі повідомлень. Основними функціями бібліотеки є:

- setBitrate() — виконує ініціалізацію контролера, встановлює швидкість передачі даних у шині (типово 100 або 500 кбіт/с для мереж ТЗ) та частоту зовнішнього кварцового резонатора модуля;
- setFilterMask() та setFilter() — конфігурують апаратні маски та фільтри ідентифікаторів повідомлень, що дозволяє на апаратному рівні відсікати кадри, які не стосуються моніторингу двигуна чи паливної системи, запобігаючи переповненню програмного буфера;
- readMessage() — зчитує вміст буфера прийому контролера та визначає довжину отриманого кадру.

Для забезпечення паралельного функціонування двох контролерів MCP2515 в програмі було створено два незалежних екземпляри класу з явним зазначенням відповідних пінів CS.

Організація обміну даними побудована за подійно-орієнтованим принципом. При надходженні в будь-яку шину кадру, який відповідає налаштованим фільтрам, відповідний модуль MCP2515 генерує низький рівень на одному з виводів INT МК. Мікроконтролер обробляє зовнішнє переривання, та виставляє прапорець готовності модуля до передачі даних. В основному циклу програми перевіряється, чи встановлений прапорець першого чи другого модуля, та проводиться зчитування пакета даних з відповідного модуля через інтерфейс SPI. Далі зчитані дані дешифруються за стандартними автомобільними формулами і зберігаються в пам’яті МК для виведення на РК-диспей.

Взаємодія з лінією K-Line реалізована за допомогою стандартного протоколу KWP2000 (ISO 14230). Для реалізації обміну з K-Line використано бібліотеку апаратного UART «Serial». Основні функції та організаційні кроки обміну:

- ініціалізація лінії: перед початком високошвидкісного обміну виконується процедура пробудження ЕБК, за допомогою функції pinMode() та часових затримок delay() на К-лінію програмно видається послідовність біт на наднизькій швидкості 5 Бод, ЕБК розпізнає цей сигнал і відповідає ключовими байтами конфігурації;
- перехід на робочу швидкість: після успішної ініціалізації об’єкт Serial конфігурується на стандартну швидкість 10400 бод за допомогою функції Serial.begin(10400);
- запит-відповідь: обмін будується за жорстким детермінованим принципом «Master-Slave», мікроконтролер відправляє діагностичний запит (наприклад, 0x21 0x01 — запит поточної групи параметрів), а бібліотека накопичує байти відповіді від ЕБК у програмному кільцевому буфері, звідки вони зчитуються функцією Serial.read().

Керування рідкокристалічним дисплеєм 16x2 здійснюється за допомогою базової стандартної бібліотеки LiquidCrystal, яка входить до складу Arduino IDE. Бібліотека приховує низькорівневі процеси стробування лінії Enable та поніблової передачі даних, надаючи високорівневі методи:

- LiquidCrystal lcd(RS, E, D4, D5, D6, D7) — конфігурація об’єкта із зазначенням пінів підключення;
- begin(16, 2) — ініціалізація геометрії дисплея;
- setCursor(col, row) — позиціонування знакопокажчика;
- print() — виведення символьних масивів або числових змінних.

Розробка інтерфейсу орієнтована на забезпечення ергономічності. Екран розділено на 6 постійних зон візуалізації, по 3 параметра у кожному рядку. Для зменшення довжини назв параметрів використовуються спеціальні символи, які генеруються в пам’яті дисплею на стадії ініціалізації. Оновлення текстових полів відбувається з фіксованою частотою 4 Гц (кожні 250 мс) за таймером, що виключає мерехтіння екрана та забезпечує плавність сприйняття інформації водієм.

3.2.3 Реалізація режимів роботи системи

Важливою функціональною особливістю розробленого ПЗ є спеціалізований режим роботи пристрою у якості логера даних. У цьому режимі система переходить у режим безперервного асинхронного перехоплення всіх кадрів, що циркулюють в обох автомобільних шинах CAN. За допомогою апаратного UART-інтерфейсу та інтегрованого USB-порту пристрій транслює сирий потік даних у ПК, де спеціальне ПЗ приймає потік, оброблює, та відображає у зручному для аналізу користувачем форматі. Також потік даних виводиться на РК-дисплей у вигляд послідовності байт.

Формат кадру логування, який виводиться на РК-дисплей, складається з ідентифікатора пакету, довжини пакету та 8 байт даних (CAN_ID | DLC | Data_Bytes [0..7]), які виводяться на дисплей з пробілами для зручності зчитування інформації користувачем. Під час відправлення пакету даних для обробки на ПК використовується інший формат пакету, сумісний з використованою для аналізу програмою «wxCAN-Sniffer.exe». Структура пакета у цьому випадку складається зі спеціальної стартової сигнатури, захопленого ID, розрахованого МК часового проміжку між приходами пакетів, розмір пакету даних (DLC), полей даних.

Даний режим є незамінним інструментом для зворотного інжинірингу недокументованих ідентифікаторів пакетів конкретного ТЗ, аналізу часових затримок між пакетами та фіксації спорадичних помилок.

Основним та найбільш функціонально насиченим режимом роботи ПЗ є режим бортового інформаційно-діагностичного контролера. На відміну від режиму логера, у цьому режимі система переходить до активного двонаправленого обміну даними з автомобільними шинами CAN та K-Line, здійснює інтелектуальну фільтрацію, глибокий парсинг сирих даних та керує багатовіконним інтерфейсом користувача на РК-дисплеї. При цьому трансляція даних у ПК через COM-порт не виконується для вивільнення обчислювальних ресурсів МК.

Даний режим перетворює розроблену систему на повноцінний БК, який забезпечує водія важливою експлуатаційною інформацією в режимі реального часу, автоматично аналізує технічний стан вузлів ТЗ та дозволяє оперативно проводити первинну діагностику без використання стороннього діагностичного обладнання.

ПЗ основного режиму (Додаток А) реалізує комплекс взаємопов’язаних алгоритмів: програмна фільтрація та декодування CAN-пакетів, діагностичне опитування шини K-Line, багатосторінковий інтерфейс користувача РК-дисплея.

На початку роботи системи МК налаштовує внутрішні маски та фільтри модулів MCP2515 на ігнорування надлишкового трафіку шини та захоплення лише суворо визначеного діапазону ID. У разі надходження переривання від одного з модулів, програма здійснює циклічний розбір вмісту інформаційних полів для виділення первинних параметрів ТЗ, таких як поточна швидкість автомобіля, оберти двигуна, температура, миттєва витрата палива тощо. Отримані значення проходять програмну лінеаризацію (множення на відповідні коефіцієнти згідно специфікації протоколу конкретного авто) і записуються у внутрішні глобальні змінні.

У разі перемикання віконного інтерфейсу на сторінку помилок, програмний модуль роботи з K-Line ініціює діагностичну сесію з ЕБК автомобіля за допомогою надсилання стандартизованих запитів за протоколом KWP2000 та чекає відповіді від ЕБК з переліком зафіксованих помилок або з інформацією про їх відсутність. При виявленні збережених кодів несправностей (DTC) програма формує список їхніх ідентифікаторів та передає до діагностичного буфера для подальшого відображення.

Для відображення великої кількості оброблених параметрів на обмеженій матриці екрана в ПЗ реалізовано архітектуру керування сторінками дисплея. Перемикання між екранами здійснюється за допомогою натискання кнопки керування, при чому під час опитування кнопки розрізняється коротке та довге натискання для реалізації різних функцій у різних випадках.

Структура інтерфейсу містить такі інформаційні сторінки:

- сторінка динамічних параметрів руху, стану двигуна та ефективності;
- дві сторінки для виведення накопичених показників пробігу авто та середньої витрати палива;
- діагностична сторінка (спеціалізоване вікно, куди виводяться зчитані коди помилок ЕБК у стандартному буквено-цифровому форматі з коротким описом помилки, або надпис «ОК» у випадку відсутності помилок).

3.3 Створення моделі системи в віртуальному середовищі

Етап комп’ютерного моделювання є невід’ємною частиною проєктування комп’ютерно-інтегрованих систем. Він дозволяє провести верифікацію логіки роботи розробленого ПЗ, перевірити правильність підключення периферійних модулів та усунути критичні помилки в алгоритмах до моменту монтажу фізичних компонентів на друковану плату.

3.3.1 Огляд онлайн-середовища моделювання Wokwi

Для попереднього тестування, верифікації алгоритмів функціонування та налагодження системи контролю було обрано сучасне середовище хмарного моделювання Wokwi. На відміну від класичних пакетів програмного забезпечення (Proteus або LTspice), Wokwi забезпечує точне потактове моделювання роботи МК архітектури AVR (зокрема ATmega328P, що є основою налагоджувальної плати Arduino Nano) у реальному часі безпосередньо у веб-браузері.

Середовище підтримує динамічну взаємодію компонентів, дозволяє підключати складні цифрові індикатори, логічні аналізатори, віртуальні термінали, а також надає вбудований компілятор та інтерактивний відлагоджувач коду.

Головною перевагою даного середовища є підтримка механізму Custom Chips (користувацькі мікросхеми API), що дозволяє за допомогою мови програмування C і спеціалізованого VWI-скриптингу створювати власні цифрові моделі компонентів, інтеграція яких відсутня у стандартній бібліотеці середовища. Це критично важливо для даного проєкту, оскільки базовий набір компонентів Wokwi не містить спеціалізованих автомобільних інтерфейсних мікросхем (CAN-шини та K-Line).

3.3.2 Створення віртуальної моделі в середовищі Wokwi

У середовищі моделювання було спроєктовано схему (рис. 3.8), конфігурація зв’язків якої визначена у файлі проєкту diagram.json. На макетній платі розгорнуто:

- мікроконтролер Arduino Nano;
- рідкокристалічний дисплей (РК) формату 16x2 на базі контролера HD44780, підключений за класичною 4-бітною сигнальною схемою;

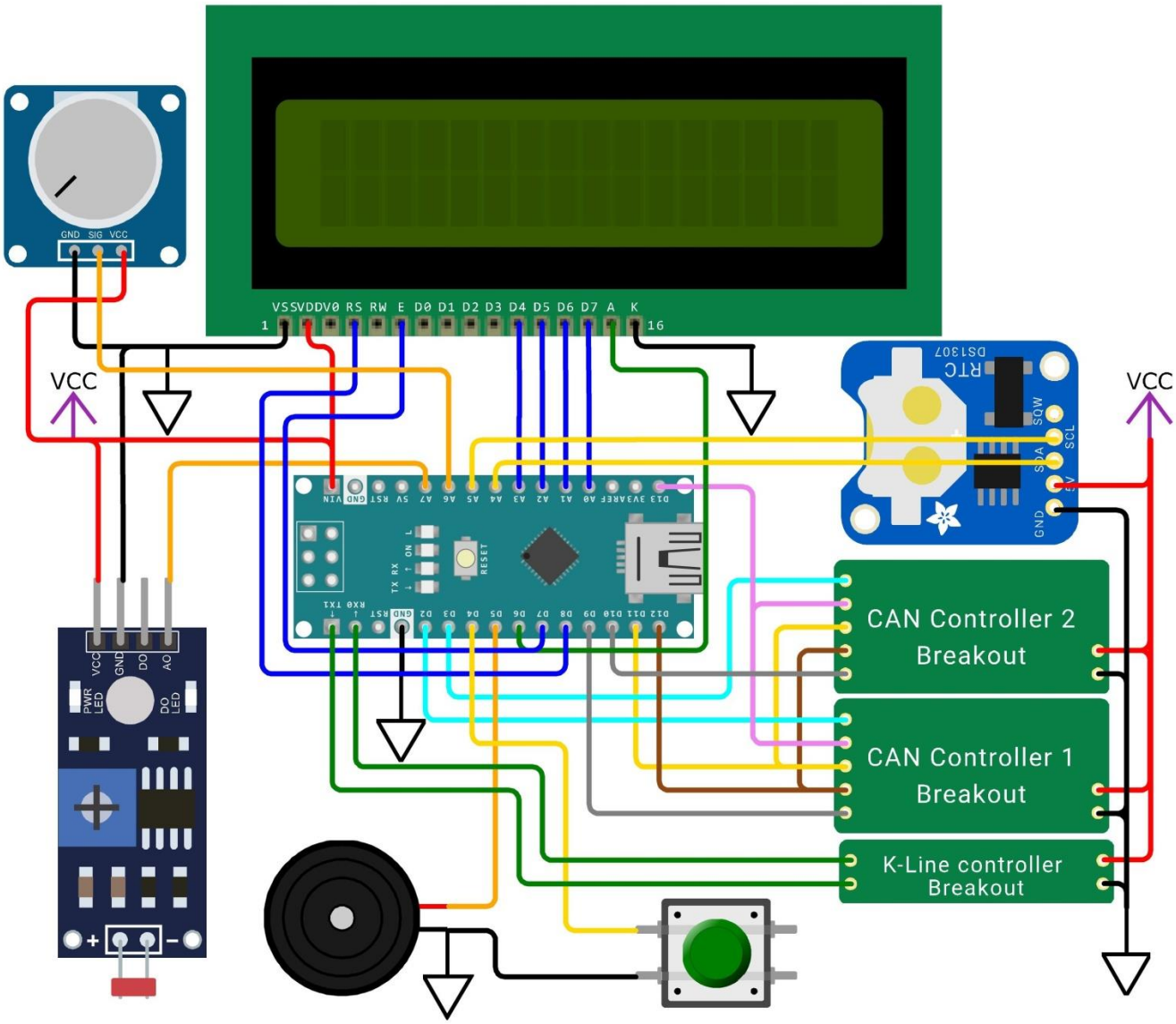


Рисунок 3.8 — Схема системи контролю в середовищі моделювання.

- два кастомні модулі для імітації роботи CAN-контролерів MCP2515, підключені паралельно до стандартних виводів шини SPI мікроконтролера, а також індивідуальними провідниками до контактів D9 та D10 (виводи Chip Select) та контактів D2 та D3 (виводи апаратних переривань МК);
- кастомний модуль для імітації фізичного рівня шини K-Line, підключений до виводів апаратного UART;
- модуль годинника реального часу, підключений до апаратних виводів I^2C;
- потенціометр, який емулює роботу резистивного поділювача напруги, та модуль фоторезистора;
- бузер для генерації звукових сповіщень та кнопку керування пристроєм.

3.3.3 Розробка кастомних чипів емуляції обміну з CAN-шиною

Оскільки в реальній схемі пристрою інтеграція з автомобільними цифровими шинами реалізована на базі двох апаратних контролерів MCP2515 (інтерфейс SPI) та компаратора LM393 (інтерфейс UART), виникла необхідність розробки відповідних кастомних чипів з програмною емуляцією роботи цих вузлів.

Кожен із двох розроблених кастомних CAN-чипів функціонує за логікою спрощеного кінцевого автомата. Оскільки повна емуляція архітектури MCP2515 є надлишковою для перевірки алгоритмів верхнього рівня, було реалізовано обмежений функціонал. У середовищі було створено новий компонент «Custom Chip», для якого був заповнений заголовковий json-файл з описом виводів чипа (рис. 3.9), при чому в список додані також порожні виводи для розподілення виводів по сторонах чипа (інформаційні виводи з лівої сторони, живлення – з правої).

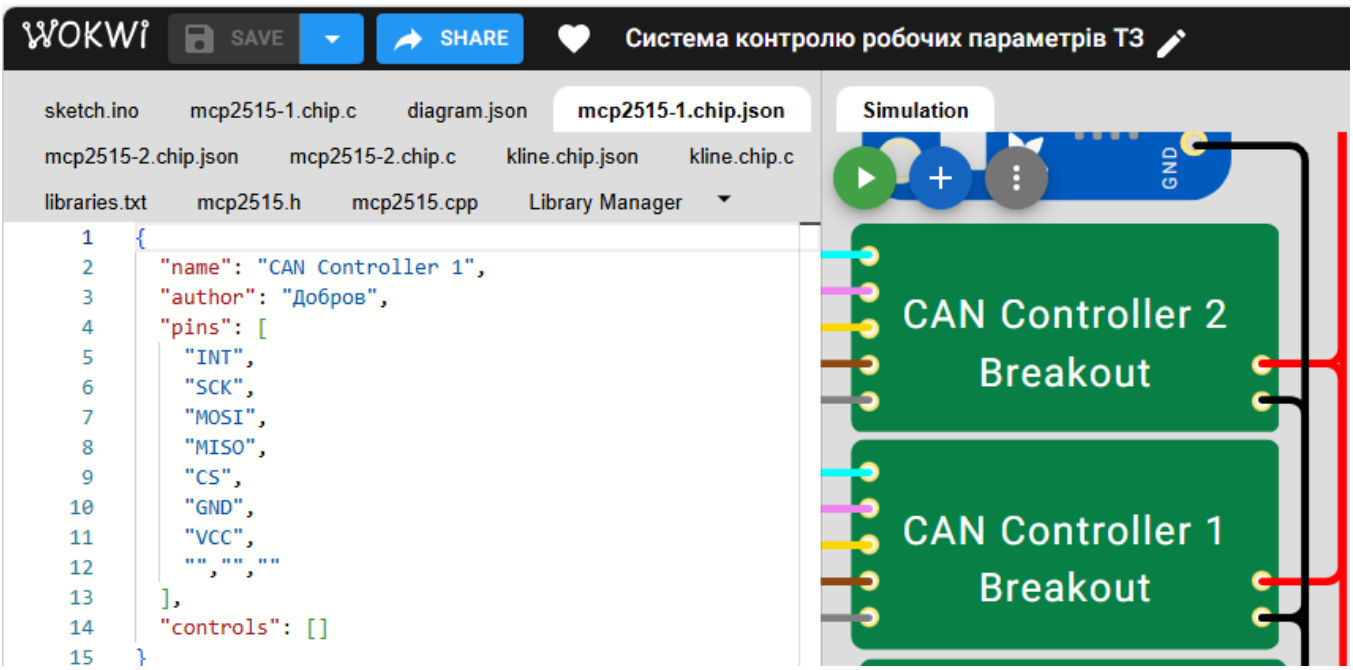


Рисунок 3.9 — Опис інформаційних виводів та виводів живлення модуля MCP2515.

Також було розроблено програму мовою С (Додаток Б), що відповідає за логіку роботи чипа, містить процедуру ініціалізації чипа з виділенням пам’яті під змінні, налаштуванням роботи виводів, встановлення процедури очікування на виводі CS, ініціалізацією таймера для емуляції генерації переривання під час надходження даних з CAN-шини. З заданою періодичністю (різною для першого та другого чипів для імітації асинхронного навантаження на шини автомобіля) чип

генерує сигнал переривання, сигналізуючи МК про наявність нового кадру. Процедура таймера місить послідовні команди встановлення високого, а потім низького рівня на виводі INT.

Взаємодія з модулем MCP2515 здійснюється за допомогою синхронного послідовного інтерфейсу SPI у режимі повного дуплексу (Full-Duplex), де кожен цикл тактування лінії SCK призводить до одночасного двостороннього обміну байтами між регістрами «майстра» та «слейва». Тож, щоб забезпечити коректну відповідь на надходження чергового байту від «майстра», «слейв» повинен «знати», який байт у відповідь необхідно буде відправити, ще після закінчення попереднього етапу обміну даними, незалежно від довжини команд, якими обмінюються пристрої. Для реалізації такого обміну в модуль логіки роботи чипа був доданий буфер на 20 байт та розроблено наступний алгоритм роботи модуля:

- під час ініціалізації обміну скиданням до низького рівня лінії CS, ще до надходження першого байту від «майстра», в буфер записується байт відповіді за замовченням (0x00);
- під час надходження першого байту даних пристрій відповідає на нього байтом за замовчуванням, записує байт, що надійшов, в буфер та виконує аналіз вхідного буфера на наявність в ньому однобайтних команд;
- аналогічна процедура виконується й під час надходження другого байту, але в цьому випадку в буфері проводиться пошук вже двобайтних команд;
- якщо якась з команд, які повинен розпізнавати пристрій для коректної емуляції обміну, знаходиться у буфері, алгоритм формує послідовність байт відповіді водночас на всі наступні кроки обміну та записує їх у буфер, починаючи з наступного за поточним елементом масиву (таким чином модуль «знає», які дані потрібно відправляти у відповідь на наступні запити від «майстра»).

Таким чином було реалізовано коректну відповідь на команду «reset», команди читання регістрів стану чипа (виконуються з модуля mcp_can перед читанням даних), команду-запит на передачу пакета даних CAN-шини. При зчитуванні чипи повертають структуровані CAN-пакети. Для першого модуля

визначено діапазон ідентифікаторів 0x1D0÷0x1D9, для другого — 0x2D0÷0x2D9. Інформаційне поле пакета (Data Bytes) заповнюється псевдовипадковими числами, що дозволяє імітувати динамічну зміну параметрів транспортного засобу. У відповідності до протоколу обміну, ідентифікатори пакетів повинні передаватися не трьома послідовними байтам, а з використанням п’яти байтів, при цьому бітові частини ідентифікатора зашифровані в різних частинах цих байтів. Для сумісності з наявним програмним забезпеченням (бібліотекою mcp_can) формування байтів ідентифікаторів пакету даних виконується за необхідним алгоритмом (рис. 3.10).

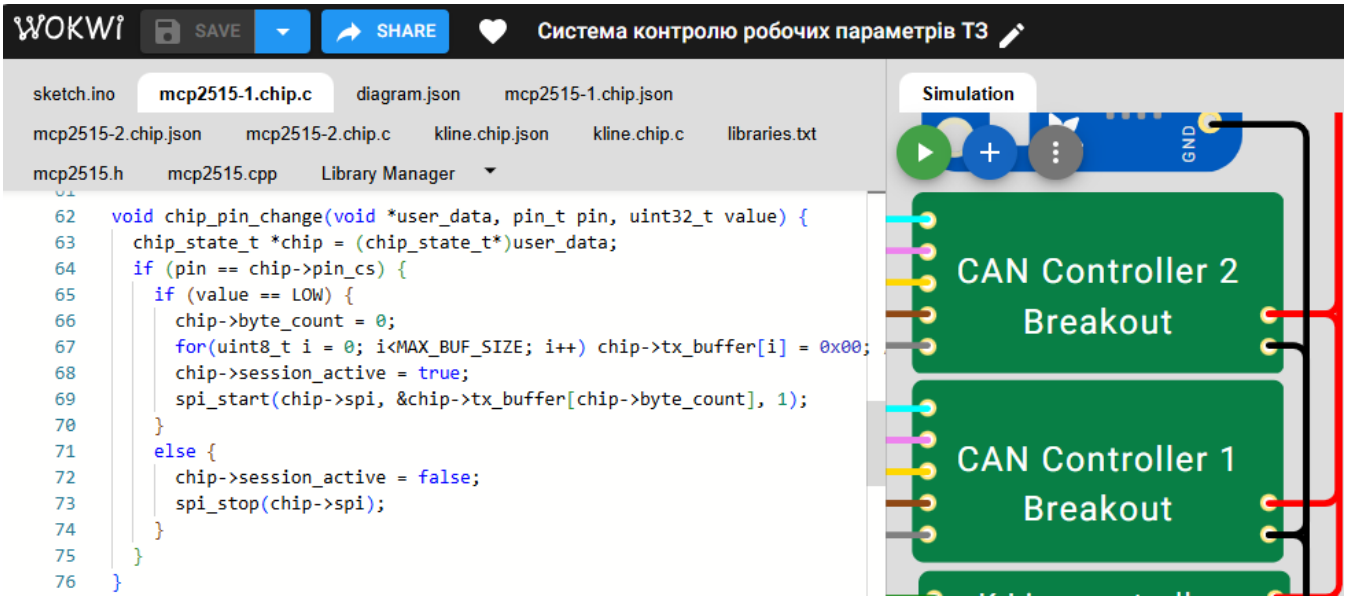


Рисунок 3.10 — Формування структури ідентифікатора CAN-пакета даних.

3.3.4 Розробка кастомного чипу емуляції обміну з K-Line-шиною

По аналогії з CAN-чипами, розроблений чип для обміну з K-Line-шиною, також функціонує за логікою спрощеного кінцевого автомата. У середовищі було додано новий компонент «Custom Chip», для якого був заповнений заголовковий json-файл з описом виводів чипа (рис. 3.11).

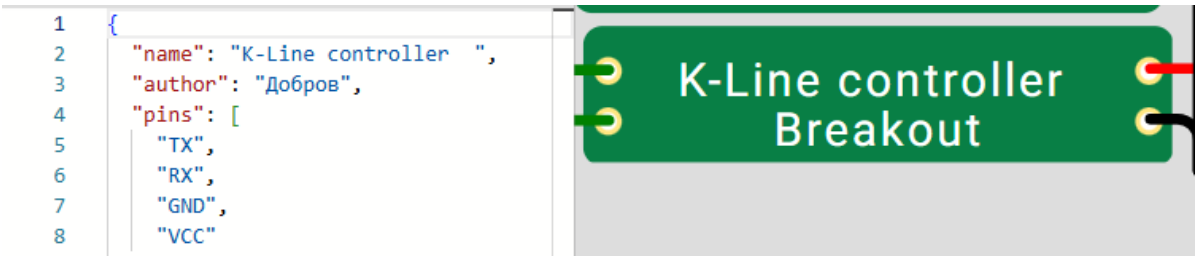


Рисунок 3.11 — Опис інформаційних виводів та виводів живлення модуля K-Line.

Також було розроблено програму мовою С (додаток В), що відповідає за логіку роботи чипа, містить процедуру ініціалізації чипа з виділенням пам’яті під змінні, налаштуванням роботи виводів, встановлення процедури очікування початку стартового біта на виводі TX, ініціалізацією внутрішнього таймера для емуляції моменту виникнення помилок.

Взаємодія з K-Line модулем здійснюється за допомогою асинхронного послідовного інтерфейсу UART у режимі напівдуплексу (Half-Duplex), оскільки в реальній K-Line шині приймач та передавач об’єднані в одну однопровідну лінію, і чип повинен «відлунювати» (ехо-бек) передані «майстром» байти назад у його приймальний буфер. Тож, щоб забезпечити коректну відповідь на надходження чергової діагностичної команди від «майстера», «слейв» емулятор ЕБК повинен аналізувати вхідний потік байтів у реальному часі, розпізнавати структуру кадру (заголовок, довжину, тип сервісу) та підраховувати контрольну суму.

Для реалізації такого обміну в модуль логіки роботи чипа був доданий буфер на 260 байт (максимально можливий розмір кадру за стандартом ISO 14230). При надходженні першого байту запиту, чип записує його в прийомний буфер та проводить розшифровку цього байта з метою визначення довжини повного пакету даних, яка зазначена в шести молодших байтах першого байту пакета. Чергові байти пакету лише фіксуються у вхідному буферу до моменту надходження останнього байту пакету. Після прийому останнього байта проводиться аналіз вмісту пакету, формування пакету-відповіді (рис. 3.12) та відправка його по лінії TX.

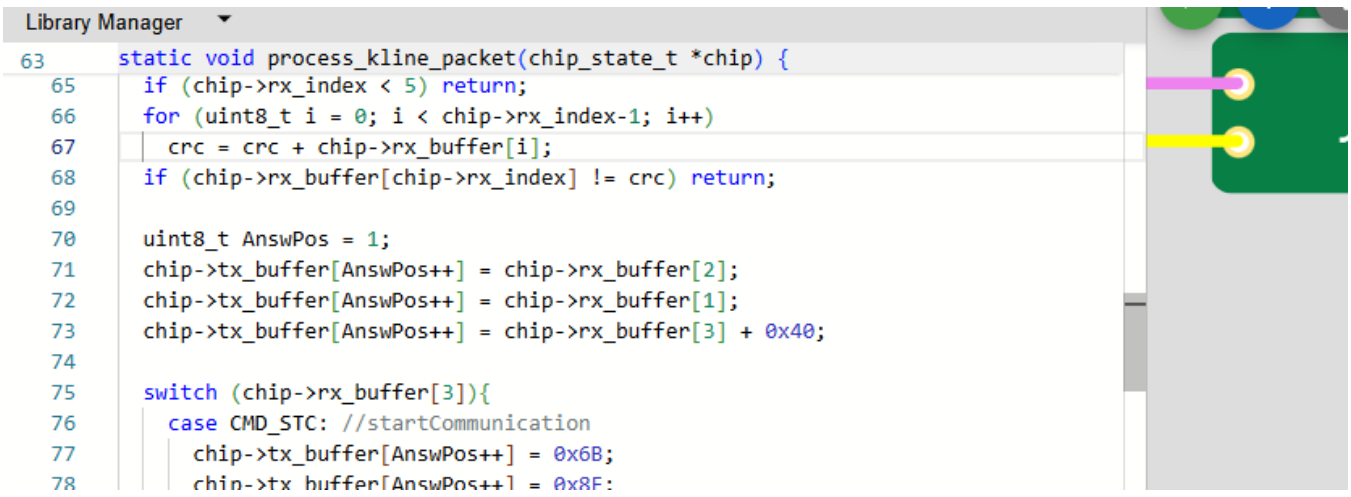


Рисунок 3.12 — Створення алгоритму формування відповіді на команді KWP2000.

Аналіз отриманого пакету полягає в наступному:

- на першому етапі розраховується контрольна сума (CRC) всіх даних вхідного пакету за виключенням останнього та порівнюється із останнім байтом пакету для визначення коректності вхідних даних;
- у другий та третій байт вихідного пакету заносяться ідентифікатори відправника та одержувача пакета із вхідного пакету (відповідні дані вхідного пакету міняються місцями;
- далі аналізується четвертий байт пакету, який згідно з протоколом обміну KWP2000 (ISO 14230) повинен містити код команди, яку потребує виконати відправник, та у відповідності до формату протоколу обміну формуються дані наступних байт вихідного пакету (один або декілька байт, які містять відповідь ЕБК);
- у випадку, коли запитується інформація про значення показників датчиків автомобіля, у відповідні байти записуються псевдовипадкові дані для імітації динамічної зміни параметрів ТЗ;
- якщо надходить команда зчитування помилок ТЗ, то в залежності від стану внутрішнього прапорця (який встановлюється в обробці переривання таймера модуля з інтервалом у 30 секунд) відправнику запиту відправляється або відповідь про відсутність помилок, або список з трьох заздалегідь визначених помилок (у цьому випадку внутрішній прапорець скидається для коректної негативної відповіді на наступний запит про наявність помилок);
- останнім етапом перед відправкою пакета виконується розрахунок довжини пакету із записуванням його до нульового байту відповіді, а також розрахунок CRC пакету із записуванням її до останнього байту пакету.

Таким чином було реалізовано емуляцію коректних відповідей на команди протоколом обміну KWP2000 (ISO 14230) з основними ідентифікаторами, що зазначені у протоколі, а також періодичну генерацію кодів несправностей та алгоритм скидання цих несправностей.

3.4 Дослідження роботи створеної віртуальної моделі

Експериментальне дослідження розробленої моделі системи контролю було проведено в середовищі віртуального моделювання Wokwi шляхом підключення створеного програмного забезпечення до імітаторів автомобільних шин обміну даними (рис. 3.13).

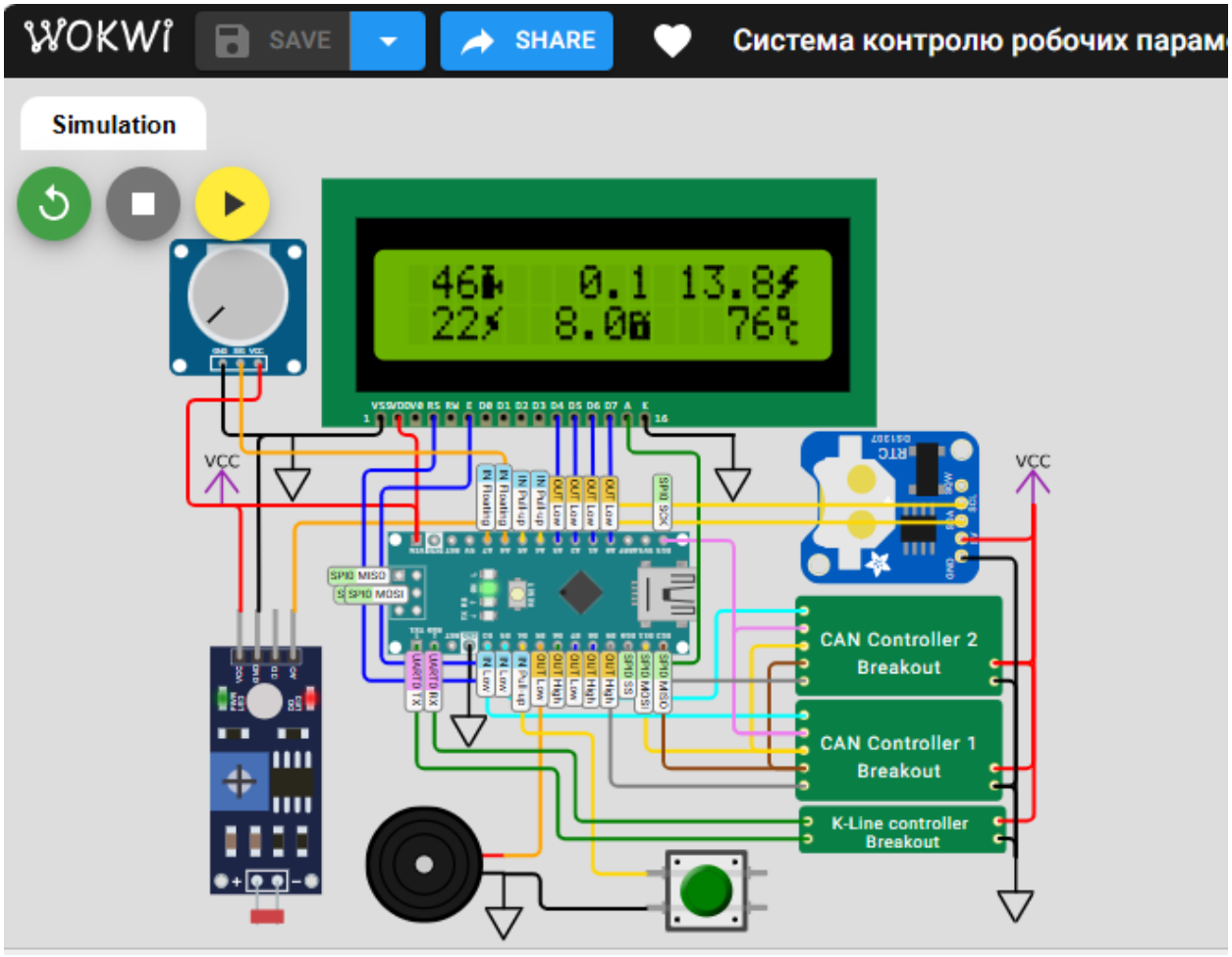


Рисунок 3.13 — Експериментальне дослідження розробленого бортового комп’ютера в середовищі віртуального моделювання Wokwi.

3.4.1 Дослідження роботи моделі в режимі моніторингу даних

Першим етапом практичного дослідження розробленої моделі став аналіз її функціонування в режимі CAN-сніффера (пасивного мережевого аналізатора). У цьому режимі система виконує паралельне захоплення пакетів з обох кастомних чипів, здійснює їх первинне парсування та передає результати на два вузли візуалізації: локальний РК-дисплей та персональний комп’ютер через віртуальний COM-порт (UART-USB інтерфейс).

Для оперативного контролю роботи пристрою було реалізовано алгоритм виведення даних на екран 16x2 (рис. 3.14). Отриманий пакет від будь-якого з CAN-модулів трансформується у текстовий рядок HEX-формату. Структура виведення має наступний вигляд: [ID] [DLC] [D0 D1 D2 D3 D4 D5 D6 D7], де ID — ідентифікатор пакета (3 символи HEX, наприклад, 1D2); DLC — довжина поля даних (1 символ, кількість байт від 0 до 8); D0..D7 — інформаційні байти, відокремлені пробілами. Через обмежену довжину рядка дисплея (16 символів) застосовано логіку автоматичного перенесення: ідентифікатор, довжина пакету та перші байти даних відображаються на першому рядку, а залишок інформаційного пакета переноситься на другий рядок РК-матриці.

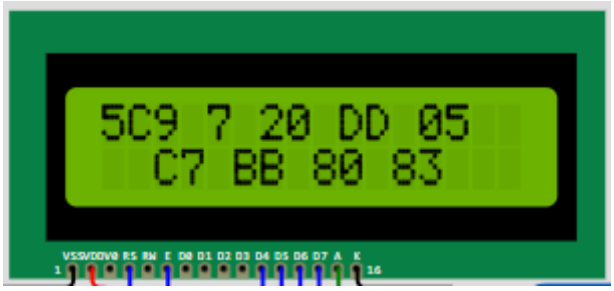


Рисунок 3.14 — Виведення пакету даних CAN-шини на дісплей 16x2.

Для можливості аналізу потоку даних мікроконтролер транслює захоплені кадри у послідовний порт. Формат кадрування був спеціально адаптований під специфікацію протоколу обміну відомої утиліти аналізу автомобільних шин даних «wxCAN-Sniffer.exe», яка використовувалась під час тестування в реальних умовах і описана нижче. Структура пакета, що відправляється в COM-порт, жорстко детермінована і не містить зайвих символів чи пробілів для мінімізації накладних витрат на передачу. Пакет складається з наступних даних:

- спеціальний стартовий байт (маркер початку кадру на ПК);
- ідентифікатор (ID) — захоплений ID з шини даних;
- розрахований мікроконтролером часовий інтервал між двома послідовними надходженнями пакетів з цим ID;
- розмір пакету даних (DLC);
- поле даних — інформаційні байти в HEX-форматі.

Така організація передачі даних у моделі повністю повторює поведінку фізичного пристрою (рис. 3.15) при його реальному підключенні до діагностичного роз'єму або панелі приладів автомобіля. Моделювання підтвердило сумісність парсера МК з протоколом wxCAN-Sniffer, що дозволяє безперешкодно перейти від віртуальних тестів до натурних випробувань у режимі моніторингу на реальному ТЗ.

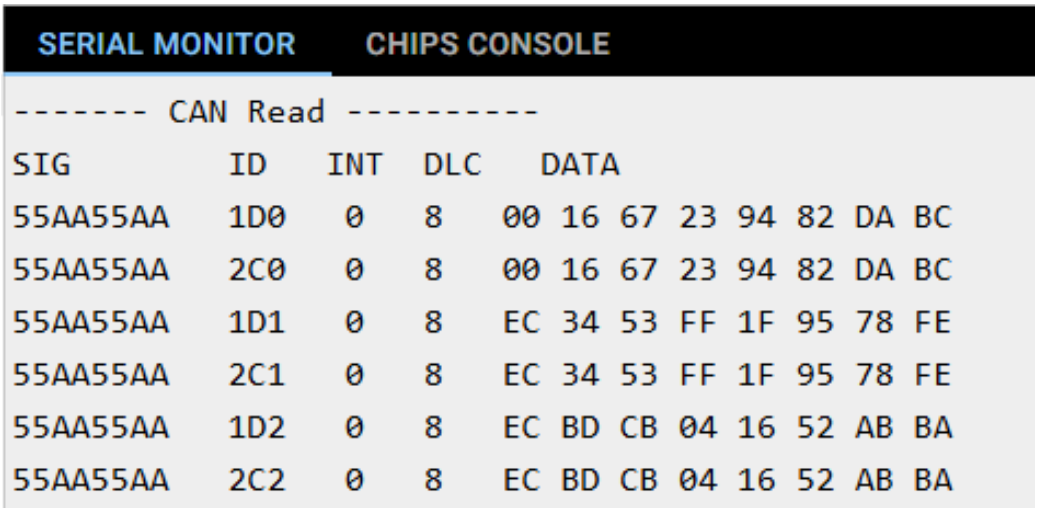


Рисунок 3.15 — Виведення пакетів даних в COM-порт ПК.

3.4.2 Дослідження роботи моделі в режимі бортового контролера

Наступним етапом практичного дослідження розробленої моделі став аналіз її функціонування в режимі БК, у якому система переходить від пасивного моніторингу до активної взаємодії з електронними компонентами ТЗ, виконуючи функції збору, інтегральної обробки, збереження в пам'ять та багаторівневого відображення експлуатаційних і діагностичних параметрів у реальному часі. Інформаційна взаємодія моделі з бортовими системами ТЗ базується на поєднанні двох інтерфейсів зв'язку, що реалізовані на рівні програмно-апаратних модулів:

- шина CAN задіяна як основний високошвидкісний канал для отримання поточних параметрів руху та роботи силового агрегату шляхом перехоплення та аналізу пакетів даних;
- діагностична лінія K-Line задіяна для виконання стандартизованих діагностичних процедур, які потребують напівдуплексного обміну за принципом «запит–відповідь» (зчитування кодів помилок та передачі команд на їх очищення з пам'яті електронного блоку керування).

Для оперативного контролю та візуалізації великого масиву параметрів на локальному РК-дисплеї було розроблено архітектуру «віконного інтерфейсу», яка складається з чотирьох послідовних інформаційних сторінок. На першій сторінці («Екран поточних параметрів», рис. 3.16) виводяться: відсоток навантаження на двигун, поточний пробіг з моменту ввімкнення з роздільною здатністю 100 м; напруга бортової мережі, відсоток відкриття дросельної заслінки, миттєва витрата палива (л/100 км у режимі руху або л/год у режимі холостого ходу), поточна температура двигуна. Ці параметри зчитуються в реальному часі через інтегровані програмні модулі CAN та перетворюються у реальні значення у відповідності до формул для конкретного ТЗ. Реалізовано одночасне виведення 6 параметрів (по 3 у два рядки) за допомогою використанням кастомних символьних іконок.

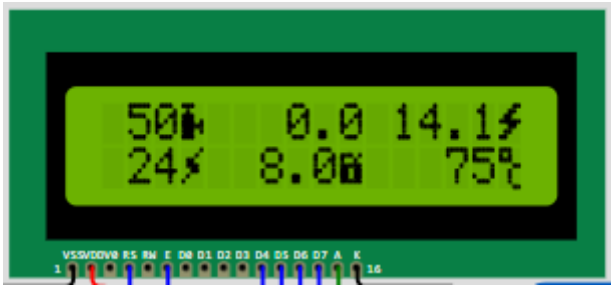


Рисунок 3.16 — Сторінка «Екран поточних параметрів» під час моделювання.

Друга та третя сторінки («Лічильники», рис. 3.17) задіяни для відображення накопичувальних лічильників: загального пробігу за весь час (км) та середньої витрати палива за цій пробіг (л/100 км), пробігу з моменту останнього ручного скидання (з точністю до 100 м) та витрати палива з моменту ручного скидання (л/100 км), другий незалежний лічильник з ручним скиданням (пробіг та витрата палива), лічильник метрів пробігу з моменту ввімкнення запалювання та середня витрата палива за поточну поїздку (л/100 км). На цих сторінках виводяться розрахункові інтегральні дані, які розраховуються програмою на підставі поточної швидкості руху миттєвої витраті пального, та періодично (з інтервалом пробігу 1 км) зберігаються в енергонезалежну пам’ять МК. Скидання значень другого та третього лічильників виконується шляхом тривалого (більше однієї секунди) натискання кнопки керування під час відображення відповідної сторінки даних на РК-диспелеї.

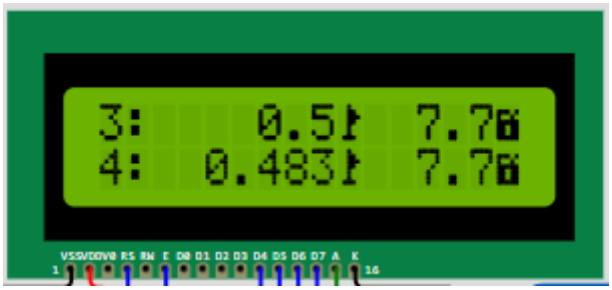


Рисунок 3.17 — Виведення поточних значень лічильниів.

Під час перемикання на четверту сторінку («Діагностичний екран DTC», рис. 3.18) виконується автоматичний запит діагностичного пакета через лінію K-Line. У випадку наявності помилок у пам’яті ЕБК на цій сторінці виводяться коди та короткий опис перших двох помилок, а в правому верхньому куті відображається загальна кількість помилок. При «довгому» натисканні кнопки керування у цьому режимі активується процедура апаратного очищення пам’яті помилок ЕБК та виконується повторний запит переліку помилок. У разі відсутності помилок, на екран виводиться напис «ОК».

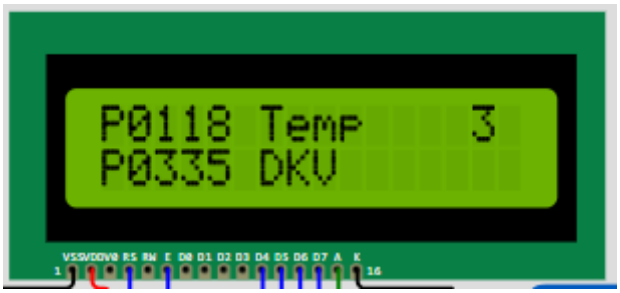


Рисунок 3.18 — Виведення зчитаних полмилок DTC під час моделювання.

Під час тестування особливу увагу було приділено перевірці стійкості роботи діагностичного модуля на сторінці помилок. На відміну від інформаційних сторінок, де запит параметрів відбувається пасивно, робота четвертої сторінки реалізує двосторонній напівдуплексний протокол зв’язку.

Віртуальне моделювання в середовищі Wokwi повністю підтвердило працездатність логіки обробки «віконного» інтерфейсу, коректність розрахунку інтегральних значень лічильників пробігу та витрати палива з їх циклічним збереженням у масиві EEPROM, а також високу надійність діагностичного модуля роботи з помилками.

3.5 Тестування роботи системи в реальних умовах

Після успішного підтвердження працездатності алгоритмів у середовищі Wokwi було виготовлено фізичний макет пристрою на базі Arduino Nano, модулів MCP2515 та компаратора LM393 для проведення повномасштабних випробувань в лабораторних умовах (за допомогою панелі приладів автомобіля) та безпосередньо на борту транспортного засобу.

3.5.1 Створення макетного зразка системи контролю

Для проведення первинних лабораторних випробувань, відпрацювання взаємодії всіх периферійних модулів та верифікації розробленого програмного забезпечення було створено діючий макетний зразок системи контролю (рис. 3.19).

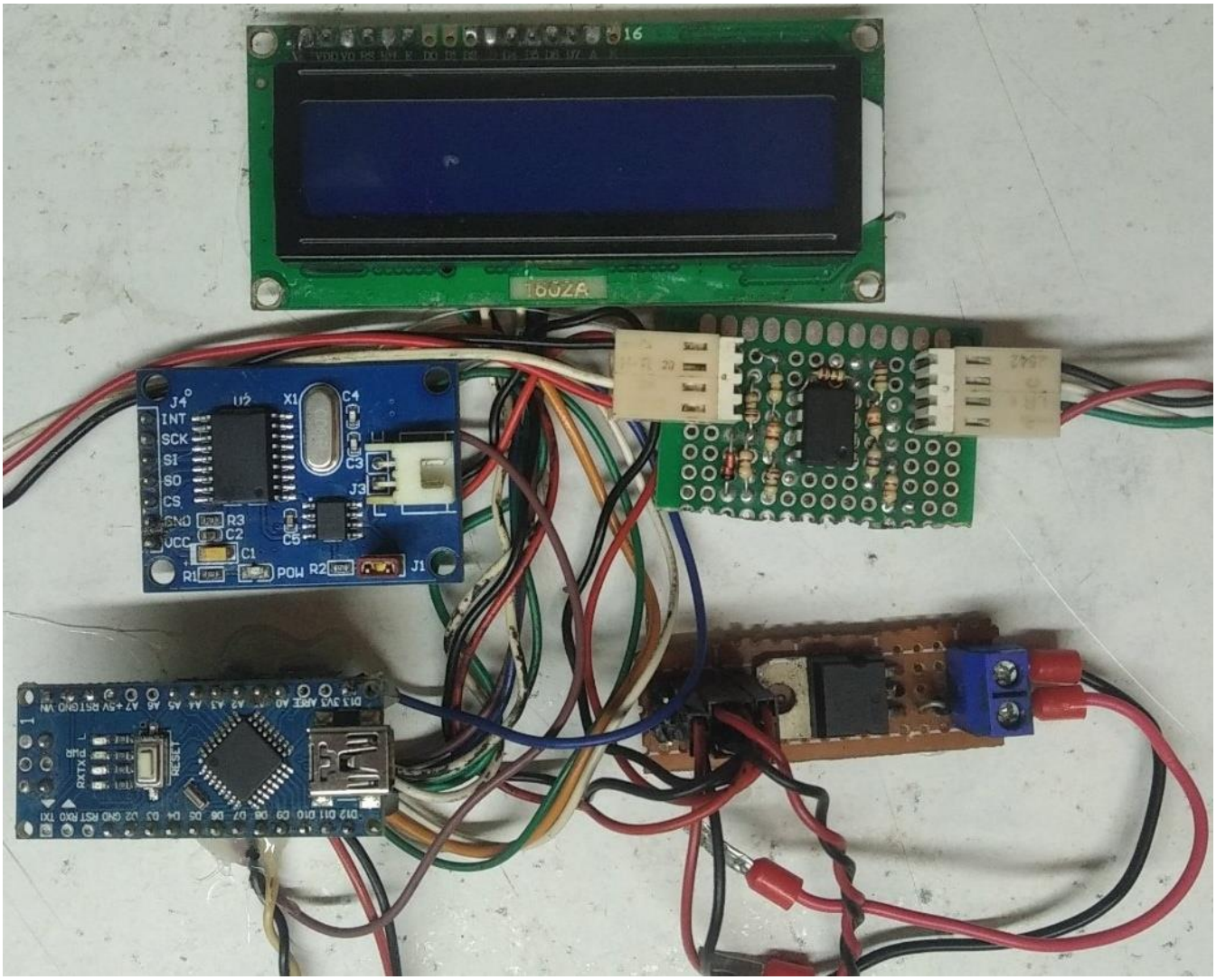


Рисунок 3.19 — Діючий макетний зразок системи контролю.

Головною метою створення макетного зразка є фізична валідація схемотехнічних рішень та перевірка працездатності системи в реальному часі до виготовлення фінальної друкованої плати. Особливістю архітектурної реалізації макетного зразка є те, що всі компоненти (налагоджувальна плата МК, модулі інтерфейсів, модуль K-Line та РК-дисплей) змонтовані на тимчасовому шасі без використання захисного корпусу. Комутацію між модулями та ланцюгами живлення здійснено за допомогою гнучких шлейфів, провідників та макетних плат.

Такий підхід на етапі макетування надає низку важливих інженерних переваг:

- гнучкість конфігурації: можливість швидкої зміни схеми підключення пінів мікроконтролера при оптимізації коду;
- доступність діагностики: безперешкодний доступ до будь-якої контрольної точки схеми для підключення вимірювального обладнання (осцилографа, логічного аналізатора) з метою аналізу форми сигналів та таймінгів;
- модульність: оперативна заміна окремих апаратних вузлів.

Водночас, представлена тестова модель є виключно проміжним (лабораторним) етапом проєктування. Наявність великої кількості провідникових з’єднань та відсутність корпусу роблять макет чутливим до механічних вібрацій, пилу та електромагнітних завад, що унеможливлює його тривалу експлуатацію безпосередньо в автомобілі. У зв’язку з цим, при переході від макета до фінального промислового зразка пристрою передбачено наступні конструктивні зміни:

- інтеграція в корпус: пристрій буде розміщено у спеціалізованому захисному корпусі (виготовленому з ударостійкого ABS-пластику методом 3D-друку), що забезпечить необхідний рівень пило- та вологозахисту (IP) та захистить компоненти від механічних пошкоджень в умовах салону авто;
- перехід на друкований монтаж: всі мікросхеми, роз’єми та пасивні компоненти будуть розведені та розпаяні на єдиній друкованій платі (PCB). Заміна провідних шлейфів плоскими печатними провідниками мінімізує внутрішній опір ліній, усуне ризик втрати контакту через вібрацію та підвищить завадостійкість пристрою.

Оскільки досліджувані транспортні засоби використовують не лише сучасну шину CAN, але й діагностичний протокол K-Line (ISO 9141 / ISO 14230), до складу пристрою інтегровано контролер, працюючий за протоколом UART, розроблений на базі прецизійного здвоєного компаратора напруги LM393 (рис. 3.20).

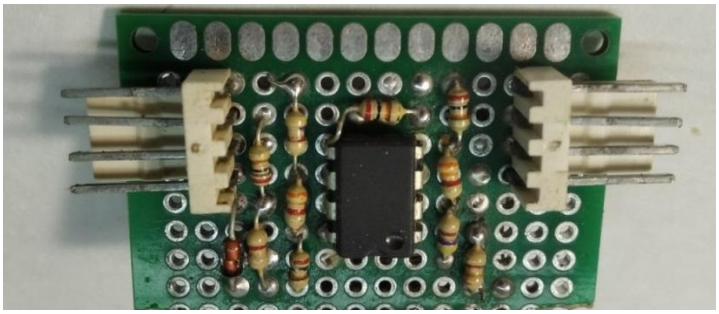


Рисунок 3.20 — Модуль зв’язку з шиною K-Line на базі компаратора LM393.

3.5.2 Опис досліджуваного транспортного засобу

Для проведення натурних випробувань та експериментального аналізу функціонування розробленого комплексу в реальних умовах експлуатації було обрано автомобіль Volkswagen Bora 2002 року випуску, побудований на платформі VAG PQ34, оснащений бензиновим чотирициліндровим двигуном об’ємом 1,6 літра з кодом модифікації AVU (потужність 102 к.с.), який керується ЕБК Simos 3.3.

Вибір даного автомобіля як об’єкта дослідження є обґрунтованим з інженерної точки зору, оскільки архітектура його бортової цифрової мережі повністю відповідає технічним вимогам та функціональним можливостям проєктованого контролера. Ключовою особливістю побудови системи обміну даними у цьому ТЗ є наявність двох незалежних шин CAN, що функціонують паралельно:

- шина CAN-Bus двигуна (High-Speed CAN) працює на швидкості 500 кбіт/с, об’єднує критично важливі блоки керування (ЕБК двигуна, блок ABS/ESP тощо), характеризується високою частотою трансляції пакетів і служить джерелом динамічних параметрів у реальному часі;
- шина CAN-Bus комфорту та мультимедіа (Low-Speed CAN) функціонує на швидкості 100 кбіт/с і зв’язує блок керування систем комфорту, модулі дверей, подушки безпеки та комбінацію приладів, та забезпечує передачу сервісних даних, параметрів пробігу, іншої допоміжної інформації.

Оскільки в розробленій моделі контролера задіяно два незалежні чипи MCP2515, архітектура цього автомобіля дозволяє повноцінно протестувати режим пасивного паралельного перехоплення даних з обох різношвидкісних мереж. Крім того, бортова система автомобіля містить стандартизовану діагностичну лінію K-Line, через яку здійснюється зв’язок з ЕБК двигуна для виконання сервісних процедур. Це дозволило в повному обсязі дослідити та верифікувати розроблений алгоритм активного діагностичного опитування, зчитування зафіксованих кодів помилок (DTC) та ініціалізації команди на їх очищення бортовим контролером.

Таким чином, комбінація двох ізольованих CAN-шин та класичної K-Line лінії в структурі Volkswagen Bora (AVU) робить його ідеальним полігоном для комплексної перевірки всіх закладених режимів роботи розробленого пристрою.

3.5.3 Інтеграція розробленої системи та дослідження протоколу обміну

Перед безпосередньою інтеграцією системи у постійний режим роботи було проведено попереднє дослідження структури реального протоколу обміну даними в шині CAN досліджуваного ТЗ. Для цього пристрій було переведено у режим логера даних (описаний у п. 3.2.3) та підключено безпосередньо до панелі приладів авто в лабораторних умовах (рис. 3.21). В подальшому пристрій було підключено до двох CAN-шин безпосередньо в авто та також було проведено легування та аналіз даних під час стоянки та руху ТЗ.

Важливою конструктивною особливістю архітектури бортової мережі досліджуваного автомобіля є те, що жодна з двох згаданих CAN-шин не виведена безпосередньо на контакти діагностичного роз’єму OBD-II. Натомість зв’язок між цими ізольованими мережами та діагностичним роз’ємом здійснюється через спеціалізований шлюз (Gateway), який у даному поколінні автомобілів інтегрований у комбінацію приладів. Цей шлюз функціонує як комутатор та апаратний фільтр трафіку: транслює на діагностичний роз’єм лише суворо обмежені діагностичні запити й повністю блокує трансляцію внутрішнього «сирого» потоку пакетів обох шин, що унеможливлює пасивне прослуховування мережевого трафіку автомобіля у разі стандартного підключення розробленого пристрою до штатного виходу OBD-II.

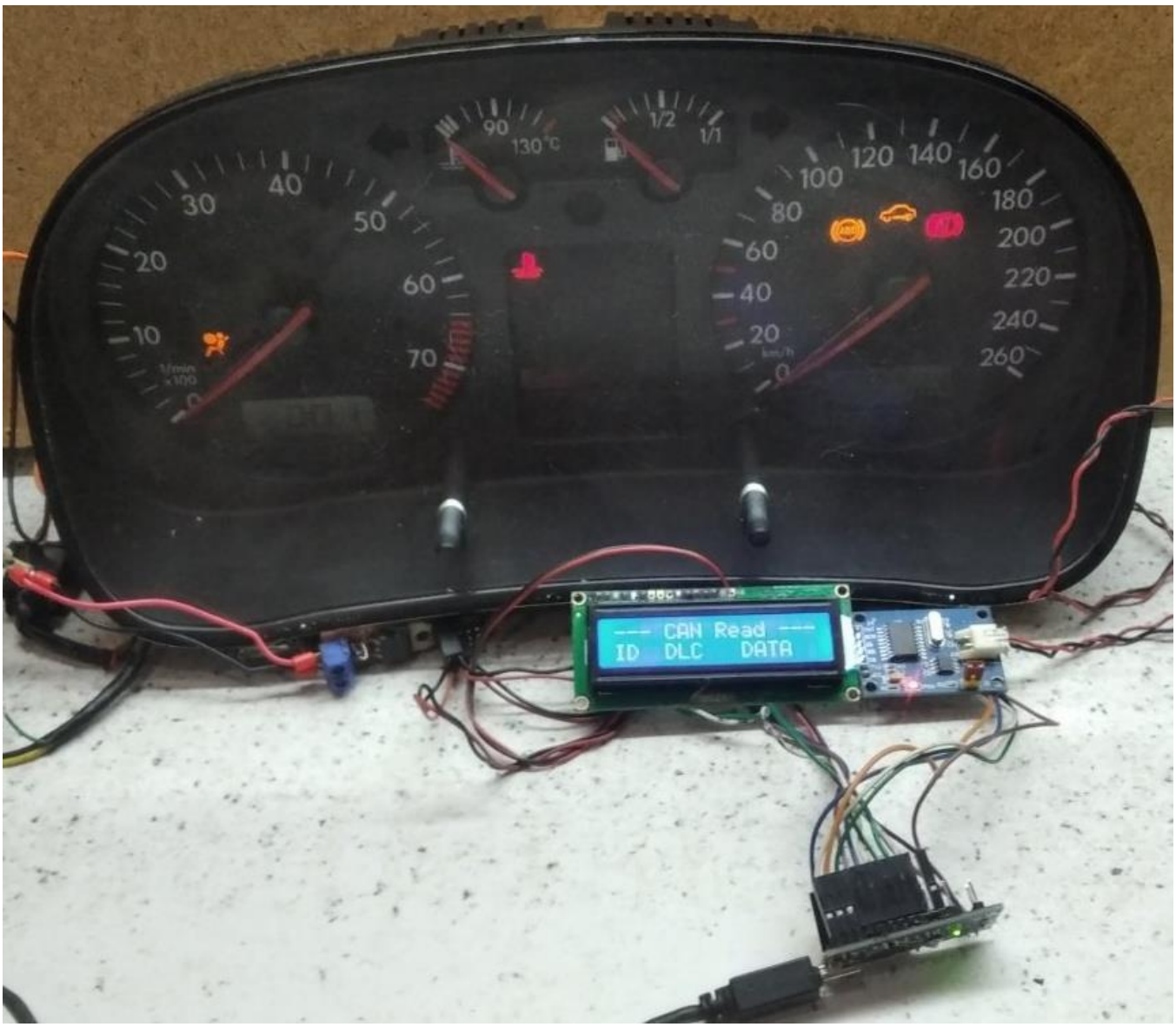


Рисунок 3.21 — Логування та аналіз протоколу обміну комбінації приладів.

Для подолання цього обмеження, забезпечення можливості проведення експериментальних досліджень та подальшої стаціонарної експлуатації пристрою від точок фізичного залягання шин (у районі панелі приладів) було підведено дві незалежні виті пари дротів, які були виведені у салон автомобіля та розміщені у безпосередній близькості до штатного роз’єму OBD-II. Така модернізація дозволила підключити розроблений прилад безпосередньо до фізичного рівня обох шин даних, забезпечивши повноцінний доступ як до активного діагностичного опитування по лінії K-Line, так і до перехоплення CAN-пакетів автомобіля в реальному часі.

Для аналізу захоплених даних та структури пакетів бортової мережі ТЗ було застосовано спеціалізоване ПЗ «wxCAN-Sniffer», головною перевагою якого є

функція динамічної колірної індикації змін у полі даних пакетів, що зумовило її вибір для проведення досліджень. Програма в реальному часі відстежує частоту оновлення кожного окремого байта за конкретними ідентифікаторами та підсвічує їх різними кольорами залежно від динаміки зміни значень. Це дозволило візуально виділити та локалізувати у щільному потоці broadcast-трафіку саме ті інформаційні кадри й розряди, які відповідають за динамічні параметри автомобіля (поточні оберти двигуна, швидкість, положення педалі газу), значно спростивши процес зворотного інжинірингу (реверс-інжинірингу) закритих протоколів автовиробника.

Трансляція перехоплених мікроконтролером кадрів у послідовний COM-порт персонального комп'ютера у строго детермінованому HEX-форматі дозволила забезпечити повну сумісність із цим аналізатором у режимі реального часу. Натурні випробування та практичне використання даного ПЗ (рис. 3.22) підтвердили високу ефективність обраного підходу для декодування внутрішньосистемного трафіку ТЗ.

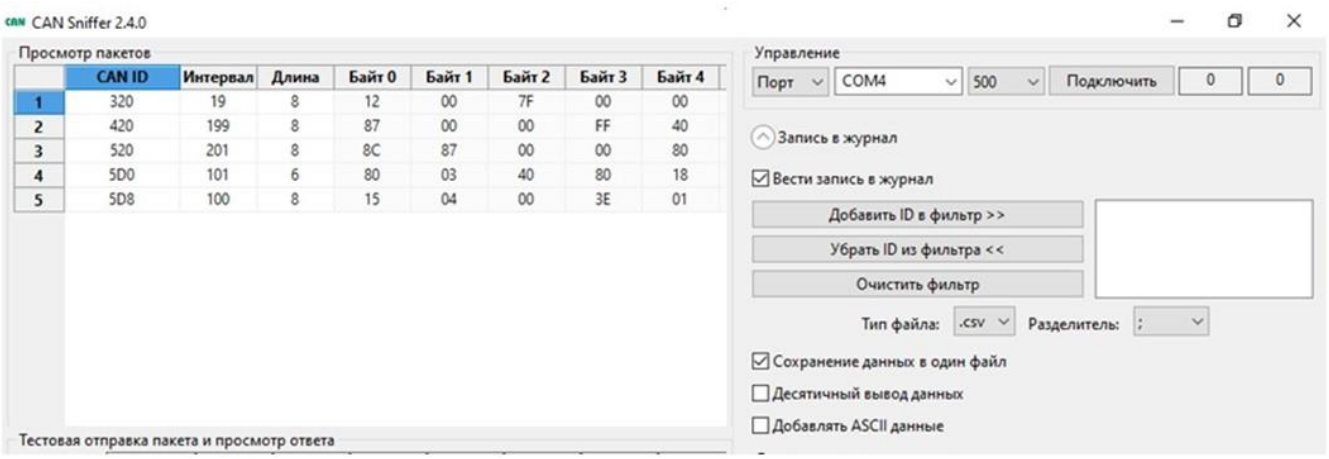


Рисунок 3.22 — Інтерфейс програми wxCAN-Sniffer під час аналізу трафіку ТЗ.

Під час роботи ТЗ на різних режимах було перехоплено та проаналізовано структуру трафіку. Шляхом зіставлення динаміки зміни байт із реальними діями (натискання на педаль газу тощо) було верифіковано ключові ідентифікатори, наприклад, кадр з ID 0x1F0 містить дані про частоту обертання колінчастого валу (об'єднані молодший та старший байти, поділені на коефіцієнт 4), а кадр з ID 0x280 містить інформацію про лінійну швидкість автомобіля та стан педалі гальма.

Дане дослідження дозволило уточнити масштабні коефіцієнти у фінальній прошивці системи для досягнення абсолютної точності відображення.

3.5.4 Дослідження роботи системи в різних сценаріях

Натурні випробування макетного зразка системи контролю проводилися у три послідовні етапи в умовах реальних поїздок, що дозволило комплексно оцінити апаратну надійність пристрою, точність обчислювальних алгоритмів та стабільність взаємодії з бортовими інтегральними структурами ТЗ (рис. 3.23).

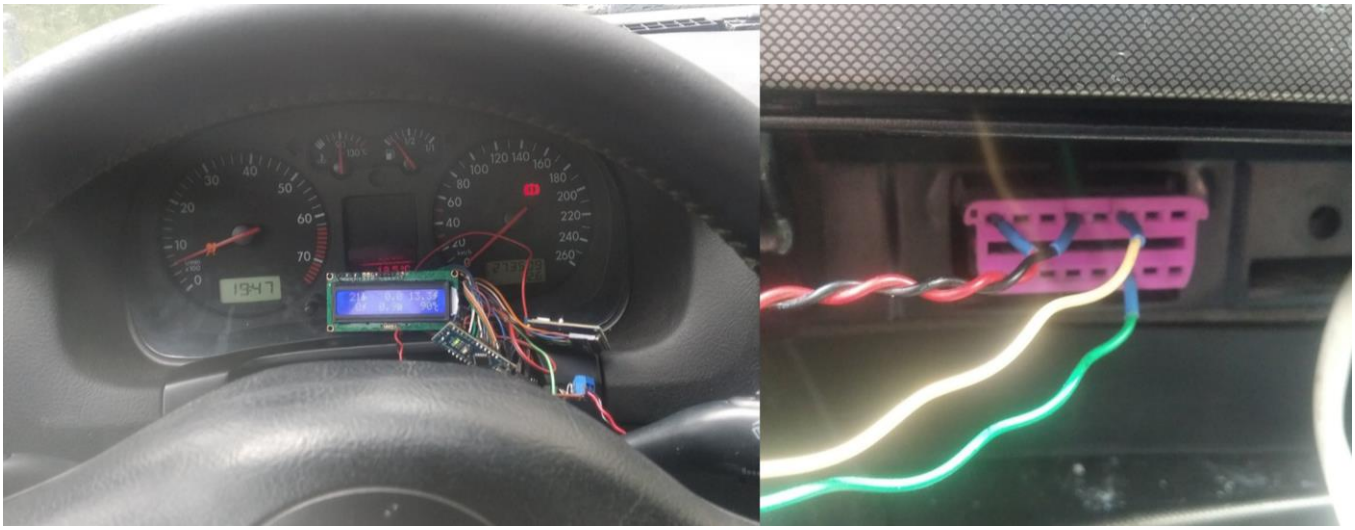


Рисунок 3.23 — Інтеграція макетного зразка у досліджувальний транспортний засіб.

На першому етапі досліджувався вплив перехідних процесів у системі електроживлення ТЗ під час пуску двигуна (рис. 3.24). У момент прокручування колінвала стартером напруга в бортовій мережі короткочасно впала до критичного значення 9.8 В, але завдяки правильно розрахованим параметрам вхідного LC-фільтра, обчислювальне ядро МК та РК-дисплей відпрацювали без перезавантажень, апаратних збоїв. На екрані пристрою миттєво й коректно відобразився пусковий стрибок обертів від 0 до 1200 об/хв, отриманий із кадру CAN-шини з ID 0x1F0.

Рисунок 3.24 — Дослідження роботи системи під час холостого ходу.

Другий етап випробувань був присвячений валідації точності розрахунку динамічних параметрів та оцінці завадостійкості інтерфейсів під час тривалого руху. У міру прогріву та руху автомобіля система чітко реєструвала зміну температури охолоджувальної рідини (від початкових 40 °C до робочих 105 °C на цьому типу двигунів), дані оновлювалися без затримок. Реалізована 4-бітна схема керування РК-дисплеєм продемонструвала відсутність спотворення символів або появи «сміттєвих» артефактів під впливом вібрацій та електромагнітних перешкод. Розрахунок миттєвої витрати палива корелював із кутом відкриття дросельної заслінки, підтверджуючи адекватність програмної лінеаризації даних (рис. 3.25). Пристрій одночасно здійснював асинхронне зчитування кадрів CAN та запит наявності помилок через модуль K-Line. Програмні кільцеві буфери та механізм переривань забезпечили повну обробку пакетів без втрати кадрів та зависання МК.

Рисунок 3.25 — Дослідження роботи системи контролю у процесі тривалого руху.

На третьому етапі натурних тестів було проведено перевірку математичного модуля інтегральних обчислень, алгоритмів роботи з енергонезалежною пам’яттю. У ході тестового пробігу було верифіковано точність накопичення даних лічильників пробігу та витрати пального. Для перевірки надійності збереження інформації було емульовано кілька циклів штатного та аварійного вимкнення

живлення контролера. Експериментально підтверджено (рис. 3.26), що при знеструмленні системи поточні значення накопиченого пробігу та сумарної витрати палива успішно фіксуються в зарезервованих комірках EEPROM, а при наступному ввімкненні пристрій коректно зчитував дані з пам’яті та продовжував обчислення без втрати накопиченої історії.

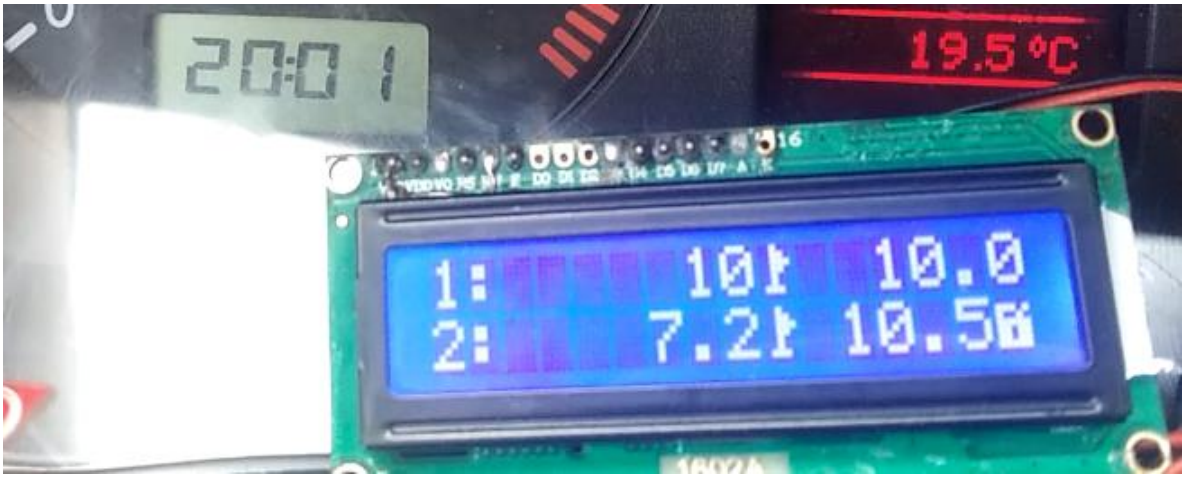


Рисунок 3.26 — Дослідження роботи алгоритмів запису в енергонезалежну пам’ять.

Додатково під час стоянки автомобіля з увімкненим запалюванням було протестовано режим активного діагностичного втручання: при переході на четверту сторінку пристрій успішно зчитав наявні сервісні коди помилок з ЕБК Simos 3.3 (рис. 3.27), а після тривалого утримання кнопки керування надіслав команду очищення переліку помилок по лінії K-Line. Позитивна відповідь від ЕБК та миттєва зміна статусу на «OK» підтвердили повну працездатність алгоритму очищення пам’яті DTC.

Рисунок 3.27 — Зчитування сервісних кодів помилок з пам’яті ЕБК.

3.6 Висновки до розділу

У третьому розділі виконано повний комплекс робіт із проєктування, програмно-апаратної реалізації, комп'ютерного моделювання та натурних випробувань системи моніторингу.

Обґрунтовано та реалізовано універсальну апаратну архітектуру пристрою на базі МК ATmega328P. Інженерне рішення із паралельного підключення двох контролерів до єдиної апаратної шини SPI з розділенням через лінії CS та обробкою асинхронних переривань INT забезпечило технічну можливість одночасного моніторингу двох ізольованих контурів ТЗ.

Розроблено та практично верифіковано схему фізичного інтерфейсу K-Line на базі компаратора LM393, яка забезпечує надійне двоспрямоване узгодження високовольтних рівнів бортової мережі ТЗ із логічними TTL-рівнями МК у режимі напівдуплексного обміну.

Створено повний цифровий двійник системи у хмарному середовищі моделювання. Розроблено унікальні поведінкові моделі цифрових емуляторів контролера MCP2515 та ЕБК двигуна для шини K-Line. Моделювання підтвердило працездатність «віконного» інтерфейсу РК-дисплея та алгоритму керування за допомогою однієї кнопки з сепарацією коротких і тривалих натискань.

За допомогою розробленого режиму CAN-логгера та спеціалізованого ПЗ «wxCAN-Sniffer» здійснено успішний реверс-інжиніринг закритого протоколіу обміну досліджуваного автомобіля, що дозволило уточнити масштабні коефіцієнти у фінальній прошивці бортового контролера.

Під час натурних випробувань на автомобілі експериментально доведено високу стабільність та завадостійкість розробленої системи у реальних сценаріях експлуатації. Програмна архітектура кінцевого автомата та оптимізовані алгоритми EEPROM забезпечили надійне збереження інтегральних лічильників пробігу та витрати палива і стабільну реалізацію безпечного режиму зчитування та очищення кодів помилок (DTC) без використання стороннього діагностичного обладнання.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ

Метою цього розділу є аналіз умов праці під час розроблення, монтажу, налагодження та експлуатації автоматизованої системи контролю робочих параметрів транспортного засобу, а також обґрунтування заходів, спрямованих на запобігання травматизму, аварійним ситуаціям, пожежам і відмовам апаратно-програмних засобів. У даному розділі під об’єктом проєктування розуміється СК, до складу якої входять апаратні модулі збору даних, блоки стабілізації живлення від бортової мережі, засоби індикації та відповідне програмне забезпечення.

4.1 Організаційно-правові основи забезпечення безпеки праці

Правову основу забезпечення безпечних умов праці під час розроблення та експлуатації системи контролю становлять Закон України «Про охорону праці» [30], Кодекс законів про працю України [31], Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці [36], Правила пожежної безпеки в Україні [35], Вимоги безпеки та захисту здоров’я під час використання виробничого обладнання працівниками [32], Правила безпечної експлуатації електроустановок споживачів [34], а також Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями [33]. Сукупність зазначених документів визначає обов’язки щодо створення належних, безпечних і здорових умов праці, організації навчання, інструктажів, технічного контролю обладнання та запобігання дії небезпечних і шкідливих чинників.

Відповідно до вимог законодавства, організація робочого місця та проведення випробувань має мінімізувати ризик травмування працівника та пошкодження штатного обладнання. Необхідно забезпечити проведення первинного, повторного та позапланового інструктажів, а також не допускати до робіт осіб, які не пройшли відповідне навчання або не ознайомлені з інструкціями з охорони праці [30, 31, 36]. Для розроблюваної системи це означає, що її монтаж, налагодження і технічне обслуговування в ТЗ повинні виконуватись лише персоналом, який ознайомлений зі схемою підключення, вимогами електробезпеки в низьковольтних мережах, порядком перевірки каналів зв’язку та правилами реагування на відмову системи.

Особливе значення для об’єкта проєктування мають вимоги, що стосуються безпечного використання обладнання та експлуатації ТЗ. Під час інтеграції в автомобіль система повинна мати справні засоби ізоляції від джерел енергії, захист від КЗ, чітке маркування органів керування, а її підключення до штатних шин має виконуватися згідно з технічною документацією, без самовільної критичної зміни схеми електропроводки ТЗ або відключення захисних елементів [32, 34]. Оскільки безпека використання залежить від умов монтажу всередині салону, після встановлення або ремонту необхідно проводити технічну перевірку перед допуском системи до роботи під час руху автомобіля [32].

Оскільки частина робіт із СК (на етапі розроблення, програмування та аналізу логів шини CAN) пов’язана з ПК, монітором та іншими екранними пристроями, робоче місце розробника/оператора також повинно відповідати вимогам до ергономіки, мікроклімату та візуального навантаження [33, 38]. Необхідно вживати заходи щодо зниження стомлюваності, забезпечення зручного робочого положення та виключення надмірного навантаження на зір і опорно-руховий апарат при тривалому аналізі цифрових потоків даних [33].

Важливою складовою правового регулювання є також пожежна безпека. При використанні системи в автомобільних умовах необхідно враховувати високі ризики короткого замикання в бортовій мережі, перегріву елементів стабілізації напруги, займання ізоляції кабелів, а також можливого поширення пожежі через горючі матеріали оздоблення салону чи приладової панелі [35]. Відповідно, система має бути встановлена у місці, яке забезпечує вільний і швидкий доступ до відключення від роз’єму OBD-II, огляду, технічного обслуговування та не перешкоджає екстреній евакуації водія і пасажирів у разі аварійної ситуації.

Під час оцінки ризиків доцільно використовувати ризик-орієнтований підхід, за яким спочатку виявляються небезпеки, далі оцінюються ймовірність і наслідки їх реалізації, після чого застосовуються заходи зниження ризику за принципом пріоритету: усунення небезпеки конструктивним рішенням, технічні заходи захисту, організаційні обмеження та інструкції для персоналу [16, 11]. Для системи контролю такий підхід є доцільним, оскільки вона поєднує апаратні МК-компоненти,

безпосереднє підключення до силової та інформаційної мережі автомобіля, а отже, вимагає комплексного аналізу безпеки на всіх етапах життєвого циклу.

Отже, законодавче регулювання охорони праці для об’єкта проєктування охоплює не лише загальні обов’язки, а й конкретні вимоги до організації безпечної експлуатації автомобільної електроніки, низьковольтних мереж, робочих місць з екранними пристроями та заходів пожежної профілактики на транспорті. Це створює нормативну основу для подальшого аналізу небезпек і розробки цільових заходів їх попередження.

4.2 Характеристика об’єкта та виявлення потенційних небезпек

Умови праці під час використання системи контролю залежать від етапу її застосування. На етапі розроблення, програмування та лабораторного налагодження розробник виконує роботи за комп’ютером, здійснює підключення модулів, плат та дисплея, перевіряє коректність сигналів, проводить тестування та аналізує результати логування. На етапі монтажу та експлуатації безпосередньо в автомобілі водій (або оператор-дослідник) додатково контактує з елементами салону, штатним діагностичним роз’ємом, підпанельною проводкою, а також зазнає впливу факторів дорожнього руху: шумів, вібрацій та мінливих параметрів мікроклімату в кабіні ТЗ.

З урахуванням особливостей об’єкта проєктування основні небезпечні та шкідливі фактори доцільно поділити на фізичні, пожежо- та електронебезпечні, психофізіологічні та організаційні. До фізичних факторів належать електричний струм бортової мережі, імпульсні та електромагнітні завади від системи запалювання, статична електрика, нагріті елементи стабілізаторів напруги, шум і вібрація під час руху ТЗ, а також несприятливі параметри мікроклімату в салоні в зимовий або літній періоди. До психофізіологічних факторів — тривала робота з екранними пристроями, статичне навантаження водія в кріслі, висока концентрація уваги під час стеження за параметрами в русі та ризик помилки через втому. До організаційних факторів належать неправильна послідовність підключення до діагностичного роз’єму, відсутність попереднього огляду ліній, неналежне маркування дротів, недостатній контроль технічного стану системи під час ходових випробувань.

Значущою небезпекою є можливість ураження електричним струмом або виникнення дугового спалаху при замиканні під час підключення, перевірки або технічного обслуговування системи. Незважаючи на те, що електронні модулі пристрою працюють від низьких напруг (5 В та 12 В), система підключається до бортової мережі ТЗ, в якій присутні потужні акумуляторні батареї з високими струмами КЗ (до кількох сотень ампер), а також можливі високольтні імпульси завад від генератора та системи запалювання двигуна. У разі пошкодження ізоляції, неправильного підключення роз'ємів, порушення вимог до з'єднання з «масою» виникає небезпека виходу з ладу не тільки пристрою, а й штатних ЕБК автомобіля.

Іншою важливою групою небезпек є пожежна небезпека в автомобілі. Короткі замикання через переплутування полярності, перевищення допустимого струму пристрою, перегрів лінійних стабілізаторів напруги через високу різницю потенціалів між бортовою мережею і логічними 5 В, неякісні контакти в місцях підключення можуть призвести до локального перегріву під приладовою панеллю, плавлення ізоляції та займання [35, 32, 34]. Для системи, яка працює в безперервному режимі, така небезпека є особливо актуальною, оскільки термічна відмова вузлів може виникнути внаслідок тривалого теплового навантаження в замкненому просторі торпеди автомобіля.

Окремо слід виділити небезпеки, пов'язані з електромагнітною сумісністю та якістю сигналів ТЗ. В автомобілі поруч із СК працюють генератор, високовольтні котушки запалювання, реле та інші потужні джерела завад. Це здатне спричинити помилки реєстрації кадрів, спотворення або втрату частини даних, появу хибних результатів. Хоча така небезпека не завжди призводить безпосередньо до травми, вона може викликати відображення помилкових критичних параметрів двигуна (наприклад, хибний сигнал про нормальну температуру при реальному перегріві), що опосередковано загрожує безпеці руху та життю водія.

Суттєве значення мають також монтажно-експлуатаційні небезпеки. Під час прокладання проводів від роз'єму до корпусу пристрою можливі механічні пошкодження провідників об гострі металеві краї елементів кузова під торпедою, переплутування ліній CAN-шини, порушення полярності живлення, або випадкове

від’єднання сигнального роз’єму в процесі руху. Такі ситуації здатні спричинити повну відмову апаратури або спотворення трансляції даних по штатній шині ТЗ.

При роботі безпосередньо в салоні рухомого ТЗ виникають небезпечні ситуації, пов’язані з обмеженим простором для монтажу, наявністю постійних низькочастотних вібрацій, тряски, підвищеної запиленості підпанельного простору. Якщо налагодження або зчитування даних ведеться водієм під час керування автомобілем у щільному міському потоці, ризик виникнення ДТП додатково зростає. Тому при аналізі небезпек важливо враховувати специфіку експлуатації системи всередині рухомого ТЗ.

До психофізіологічних небезпек належать інформаційне перевантаження, зорове напруження через зчитування інформації з невеликого дисплея, тривале перебування водія в статичній позі в автомобільному кріслі, а також ризик помилок та відволікання від керування ТЗ. Для користувача системи особливо небезпечними є ситуації, коли інтерфейс екрана перевантажений або параметри відображаються нечітко. У такому разі зростає ймовірність хибних дій або запізнілого реагування.

Таким чином, до основних небезпечних ситуацій на об’єкті проєктування слід віднести: підключення пристрою до діагностичного роз’єму під напругою; пошкодження кабелів об гострі кромки деталей салону; перегрів або коротке замикання у вузлі живлення пристрою; порушення передачі даних через автомобільні завади; відволікання уваги водія від дорожньої обстановки через незручний інтерфейс дисплея; переплутування полярності ліній живлення та сигнальних шин під час монтажу; вплив тривалої вібрації на надійність клемних з’єднань приладу. Перелічені небезпеки потребують кількісної оцінки ризику та розробки конкретних організаційно-технічних заходів.

4.3 Дослідження ризику реалізації потенційних небезпек на об’єкті проєктування та розробка заходів щодо їх попередження

Оцінка ризику виникнення небезпеки є важливим етапом у процесі охорони праці, який дозволяє визначити можливість виникнення небезпечних ситуацій та визначити необхідні заходи щодо їх усунення або зменшення на мінімум. Для проведення оцінки ризику можна використовувати різні методики, зокрема:

- методика Хазоп використовується для аналізу небезпечних інцидентів та визначення можливості їх виникнення з аналізом робочих процесів, ідентифікацією потенційних небезпек, визначеням ефективність існуючих заходів запобігання та розробкою пропозиції щодо усунення недоліків;
- метод оцінки ризику за допомогою матриці полягає у визначенні ймовірності виникнення небезпечної ситуації та ступеня її наслідків;
- класифікація та оцінка ризику виникнення електромагнітного випромінювання за ступенем припустимості;
- метод функціонального аналізу використовується для визначення функцій, які виконуються на робочому місці, та оцінки їх можливого впливу на здоров’я працівників.

Для оцінювання рівня ризику використано експертний матричний метод, заснований на визначенні ймовірності реалізації небезпеки P та тяжкості її наслідків S. Інтегральний показник ризику визначається як добуток:

$$R = P \cdot S,$$

де P — бал імовірності виникнення небезпечної події; S — бал тяжкості наслідків. Для уніфікації оцінювання приймається п’ятибальна шкала, наведена в таблиці 4.1.

Таблиця 4.1 — Шкала оцінювання імовірності та тяжкості наслідків.

Бал	Імовірність виникнення небезпеки	Тяжкість наслідків
1	малоймовірно, подія можлива лише за збігу несприятливих умов	незначні наслідки, короткочасний дискомфорт, відсутність пошкодження обладнання автомобіля
2	рідко, подія виникає епізодично	легкі наслідки, невелике пошкодження пристрою або коротка зупинка роботи
3	періодично, подія можлива в реальних умовах експлуатації	відчутні наслідки, тимчасова втрата працездатності або відмова окремого вузла
4	часто, небезпека реалізується за типових порушень режиму роботи	тяжкі наслідки, значне пошкодження приладу чи проводки ТЗ, тривала зупинка авто
5	дуже часто або за відсутності захисту майже неминуче	критичні наслідки, важке травмування внаслідок ДТП, пожежа в салоні автомобіля або значні втрати

Відповідно до прийнятої шкали формується матриця ризику, наведена в таблиці 4.2. Для подальшого аналізу приймаємо таку інтерпретацію: R = 1÷4 — низький ризик; R = 5÷9 — середній ризик; R = 10÷16 — значний ризик, що потребує обов'язкових заходів зниження; R = 17÷25 — високий ризик, за якого роботу без додаткових заходів безпеки проводити не допускається.

Таблиця 4.2 — Загальна матриця оцінювання ризику.

P \ S	**1**	**2**	**3**	**4**	**5**
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

Насамперед розглянемо електротехнічні й пожежні небезпеки, оскільки саме вони мають найбільшу потенційну тяжкість наслідків для системи контролю параметрів та бортової мережі ТЗ (табл. 4.3). Як видно з оцінювання, найвищого пріоритету набувають заходи електробезпеки та пожежної профілактики в ТЗ. Для їх реалізації в конструкції системи необхідно передбачити обмеження струму на входах, використання захисних діодів або TVS-діодів (супресорів) на інтерфейсних лініях CAN та K-Line для гасіння імпульсних наведень бортової мережі автомобіля, а також обов'язкове маркування живильних і сигнальних провідників. Кабелі потрібно підбирати в термостійкій ізоляції (стійкій до автомобільних масел та температурних перепадів) із достатньою механічною міцністю на згин, а пластиковий корпус приладу має бути надійно закріплений у салоні, виключаючи рух під дією вібрації ТЗ [35, 32, 34].

Для системи контролю параметрів ТЗ важливими є також ергономічні та організаційні ризики, оскільки значна частина небезпечних ситуацій у дорозі пов'язана безпосередньо з діями водія та читабельністю приладів індикації (табл. 4.4). Ризик, пов'язаний із відволіканням водія та втомою, є значним, оскільки зчитування інформації не повинно заважати стеженню за дорожнім рухом.

Таблиця 4.3 — Оцінювання електротехнічних та пожежних ризиків.

Небезпека	Причина виникнення	P	S	R	Рівень ризику	Основні заходи зниження ризику
Ураження електричним струмом	пошкодження ізоляції дротів, КЗ на масу автомобіля, невірне підключення до OBD-II	2	5	10	значний	знеструмлення перед підключенням, надійна ізоляція дротів, підключення до штатного запобіжника
Коротке замикання в автомобільній мережі	переплутування полярності живлення, помилка монтажу, перетирання кабелю	3	4	12	значний	струмове обмеження, чітке маркування ліній CAN/K-Line, контроль монтажу за схемою
Перегрів вузла стабілізації напруги пристрою	велика різниця напруг (14В->5В), тривала робота, відсутність вентиляції під торпедою	2	5	10	значний	використання імпульсних крокових перетворювачів, вентиляційні отвори, добір елементів із запасом по струму
Пожежа в автомобільних кабельних з’єднаннях	послаблення клем від вібрації, локальний нагрів роз’ємів, неякісна пайка	2	5	10	значний	використання сертифікованих авто-роз’ємів, фіксація кабелів, контроль щільності посадки в гніздо
Пошкодження компонентів електростатич-ним розрядом	монтаж плат в салоні без антистатичних заходів	3	2	6	середній	використання антистатичного браслета при збиранні, захисний корпус для пристрою

З цієї причини корпус пристрою з дисплеєм 16x2 повинен відповідати ергономічним вимогам: він має бути розміщений так, щоб водію достатньо було кинути швидкий погляд без значного повороту голови від лобового скла; відблиски від сонця мають бути усунені захисним козирком; відображення символів має бути стабільним, без миготіння [33].

Під час тривалого збору даних (режим логера CAN) аналіз отриманих файлів на ПК повинен виконуватися виключно під час стоянки автомобіля або іншим оператором-дослідником на пасажирському сидінні [33, 38].

Таблиця 4.4 — Оцінювання ергономічних та організаційних ризиків.

Небезпека	Причина виникнення	P	S	R	Рівень ризику	Основні заходи зниження ризику
Зорове перенапру-ження водія	тривале спостереження за дрібними символами екрана під час руху, сонячні відблиски	4	2	8	середній	ергономічне розміщення дисплея в зоні прямої видимості, підбір яскравості підсвітки
Статичне навантаження та втома під час поїздки	тривале перебування за кермом, незручне положення тіла	4	3	12	значний	вірне регулювання авто-крісла, перерви в русі під час тривалих поїздок
Відволікання водія через надлишок даних	виведення занадто великої кількості параметрів на екран, відсутність акценту на аваріях	3	3	9	середній	раціональна структура інтерфейсу, великий шрифт, звукове або миготливе сповіщення про перегрів двигуна
Пошкодження приладу або спотикання при посадці	хаотичне прокладання проводів в ногах водія, відсутність кріплення	3	2	6	середній	прокладання проводів під штатними картами обшивки салону, надійна фіксація стяжками
Погіршення самопочуття через мікроклімат у кабіні	тривалі поїздки в закритому салоні авто, вплив літньої спеки або завад опалення	3	3	9	середній	контроль вентиляції та кондиціонування салону ТЗ, регламентовані організаційні перерви

Окремо оцінюються ризики, що виникають під час безпосереднього монтажу, комутації та ходового налагодження системи на борту ТЗ. Результати, наведені в таблиці 4.5, показують, що для етапу ходового налагодження критично важливими є заходи з попередження самовільного руху автомобіля, перетирання дротів від вібрації та захисту входів від високих імпульсних напруг ТЗ. Перед пуском двигуна автомобіль має бути надійно зафіксований стоянковим гальмом. Будь-які маніпуляції з платою, перепрошивка або зміна клемних з’єднань повинні виконуватися виключно при вимкненому запалюванні та знеструмленому пристрої.

Таблиця 4.5 — Оцінювання ризиків під час монтажу та налагодження системи.

Небезпека	Причина виникнення	P	S	R	Рівень ризику	Основні заходи зниження ризику
Перетирання дротів системи об гострі краї деталей салону	механічний вплив, постійна вібрація і тряска, натяг лінії	3	3	9	середній	використання захисних гофрованих рукавів, правильний радіус вигину ліній
Перевищення допустимих сигналів на входах МК	неправильне підключення К-лінії, відсутність узгодження рівнів	3	4	12	значний	точне застосування дільників напруги, перевірка рівнів сигналів
Термічний опік об гарячі вузли під капотом автомобіля	підключення або огляд датчиків відразу після тривалої поїздки	2	4	8	середній	виконання монтажних робіт у підкапотному просторі тільки після охолодження двигуна
Раптовий рух автомобіля під час налагодження	пуск двигуна при ввімкненій передачі під час зняття осцилограм або підключення плат	2	5	10	значний	обов’язкове встанов-лення автомобіля на стоянкове гальмо, блокування коліс упорами
Втрата логів CAN-шини або збій зв’язку по K-Line	нестійкий контакт у гнізді OBD-II через вібрацію, програмні помилки	3	3	9	середній	використання якісних штекерів із фіксацією, логування помилок зв’язку в програмі

На підставі проведеної оцінки можна зробити висновок, що найбільш суттєвими для об’єкта проєктування є такі ризики: коротке замикання в бортовій мережі, термічний перегрів автомобільних дротів або стабілізаторів живлення, неправильне узгодження рівнів сигналів, відволікання уваги водія в русі та ризик травмування при незафіксованому автомобілі під час пуску ДВЗ. Саме на зниження цих ризиків повинні бути спрямовані першочергові заходи.

До технічних заходів попередження небезпек належать: застосування DC-DC стабілізаторів живлення з високим ККД та низьким тепловиділенням; встановлення плавкого запобіжника у ланцюг живлення; використання узгоджувального каскаду на компараторі із обмеженням амплітуди сигналів до 5 В; захист CAN-ліній варисторами; прокладання проводів у захисних гофротрубах; надійна пайка елементів для протидії руйнуванню від вібрації; використання надійного корпусу, що виключає контакт рук водія зі струмовідними доріжками плати [32, 34, 16].

До організаційних заходів слід віднести: проведення інструктажів перед початком монтажу в салоні ТЗ; допуск до ходових випробувань лише осіб, які мають посвідчення водія відповідної категорії та ознайомлені з правилами безпеки на транспорті; перевірку надійності фіксації корпусу приладу перед початком руху; ведення чек-листа перевірки з'єднань перед виїздом на дороги загального користування; регулярний огляд стану проводів у зоні педального вузла автомобіля; резервне копіювання файлів логування на зовнішні носії. Водій/оператор має бути чітко поінформований про залишкові ризики і порядок дій у разі появи диму чи запаху горілої ізоляції в салоні [30, 31, 36, 35, 32].

Правила безпечної експлуатації системи контролю повинні включати такі обов'язкові вимоги. Перед кожним виїздом ТЗ слід проводити зовнішній огляд корпусу приладу, перевіряти щільність посадки роз'єму OBD-II, цілісність гофрованих рукавів. Підключення та перепідключення сигнальних ліній або інтерфейсу SPI/UART необхідно виконувати тільки при вимкненому запалюванні ТЗ. Не допускається експлуатація пристрою при появі запаху перегріву, нестабільності виведення даних на дисплей чи видимих пошкодженнях ізоляції.

Під час руху автомобіля водію категорично забороняється проводити будь-яке програмне налагодження, змінювати конфігурацію чи налаштовувати елементи плати; всі доопрацювання коду та перезапуск пристрою мають виконуватися строго на парковці при зупиненому авто [33]. Кабелі слід фіксувати під торпедою так, щоб повністю виключити їх провисання в зону ніг водія, а також контакт з гарячими патрубками обігрівача салону або рухомими елементами рульового вала. Ходові випробування в складних дорожніх чи погодних умовах (ожеледь, злива) мають бути припинені для забезпечення безпеки екіпажу ТЗ [32, 34].

Для зниження психофізіологічного навантаження на водія реалізовано ергономічну побудову інтерфейсу дисплея: структура екрана виключає дрібні текстові блоки, оновлення значень RPM та температури оптимізовано за частотою (4 Гц) для усунення стомлюючого миготіння, а при досягненні критичних обертів або перегріві двигуна передбачено динамічне попередження на екрані, що полегшує сприйняття інформації в дорозі [33, 38].

Отже, застосування наведених організаційних і технічних заходів дозволяє знизити ризик реалізації найбільш небезпечних сценаріїв в автомобілі до прийнятного, безпечного рівня. При цьому пріоритетними є правильне проєктування схеми підключення, наявність засобів плавкого і завадового захисту, унеможливлення відволікання водія від керування та суворе дотримання регламенту ходових випробувань.

4.4 Висновки до розділу

У розділі проведено аналіз умов праці під час розроблення, монтажу, налагодження та експлуатації автоматизованої системи контролю параметрів транспортного засобу. Встановлено, що найбільш суттєвими небезпеками для водія та бортової електроніки автомобіля є коротке замикання в низьковольтній мережі ТЗ, термічний перегрів вузлів стабілізації живлення плати, пошкодження дротів через автомобільні вібрації, відволікання уваги водія від дорожньої обстановки при зчитуванні даних з екрана, а також специфічні ризики (раптовий рух авто, опіки), що виникають під час монтажу та тестування безпосередньо на борту ТЗ.

На основі матричної оцінки ризику визначено пріоритетні напрями зниження небезпек: інтеграція імпульсних крокових стабілізаторів напруги із захисними автомобільними запобіжниками, надійна вібростійка ізоляція та фіксація кабельного господарства в салоні під приладовою панеллю, раціональне ергономічне розміщення дисплея в полі зору водія без перекриття огляду дороги, а також суворе дотримання правил безпеки під час пуску ДВЗ та проведення ходових випробувань. Запропоновані заходи мають організаційний і технічний характер, охоплюють як схемотехнічні рішення, так і регламент експлуатації пристрою.

Реалізація запропонованих рішень дозволить підвищити рівень безпеки праці, мінімізувати ймовірність виникнення аварійних ситуацій чи ДТП, зберегти цілісність штатної електропроводки автомобіля та забезпечити надійність логування експериментальних даних шини CAN і лінії K-Line.

ВИСНОВКИ

У дипломній роботі вирішено актуальну науково-технічну задачу з розробки та впровадження автоматизованої системи контролю робочих параметрів ТЗ.

На основі аналізу існуючих засобів моніторингу обґрунтовано доцільність створення автономного мікроконтролерного пристрою, який поєднує пасивне прослуховування високошвидкісних потоків CAN-шин для отримання динамічних параметрів руху та напівдуплексний обмін по K-Line для сервісних процедур.

Спроєктовано модульну структуру пристрою на базі МК ATmega328P, що містить два паралельно підключені контролери MCP2515 на єдиній шині SPI для одночасного моніторингу ізольованих контурів CAN-шин, а також оригінальну завадостійку схему узгодження рівнів K-Line на базі компаратора LM393.

Успішно реалізовано концепцію цифрового двійника системи у хмарному середовищі Wokwi. Створені користувацькі C-модулі забезпечили повноцінну емуляцію апаратних протоколів (SPI, UART) та специфікацій ISO 14230 з підрахунком CRC, що дозволило повністю верифікувати логіку «віконного» інтерфейсу та алгоритм однокнопкового керування до етапу фізичного складання.

За допомогою додаткових витих пар, підключених безпосередньо до фізичних ліній CAN-шин, та підключення до стандартного роз’єму ODB-II проведено натурні випробування на автомобілі. За допомогою розробленого режиму CAN-логгера здійснено реверс-інжиніринг внутрішнього трафіку досліджуваного ТЗ, що забезпечило точне калібрування масштабних коефіцієнтів у прошивці пристрою.

Натурне тестування макетного зразка системи в реальних умовах довело його високу завадостійкість та стабільність. Верифіковано коректність розрахунку лічильників пробігу, циклічного запису інтегральних лічильників в EEPROM, а також роботу алгоритму безпечного зчитування та очищення кодів помилок (DTC).

У межах розділу з охорони праці проведено ідентифікацію потенційних небезпек та запропоновано комплекс технічних та організаційних заходів для підвищення рівня безпеки праці, мінімізації ймовірності виникнення аварійних ситуацій чи ДТП, збереження цілісності штатної проводки ТЗ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AEC - Q100 - REV-J. Failure mechanism based stress test qualification for integrated circuits in automotive applications. URL: http://www.aecouncil.com/Documents/AEC_Q100_Rev_J_Base_Document.pdf.
2. Arduino Pro Mini 328 - 5V/16MHz. URL: https://docs.arduino.cc/retired/boards/arduino-pro-mini/.
3. Arduino UNO R3 Document. URL: https://docs.arduino.cc/hardware/uno-rev3.
4. Arduino® Nano. User Manual SKU: A000005. URL: https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf.
5. Arduino® UNO R3. User Manual SKU: A000005. URL: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf.
6. Autel. MaxiCOM Ultra Lite. URL: https://autel.com/au/wp-content/themes/autel/u/cms/www/202404/01020049ifun.pdf.
7. CAN in Automation (CiA). URL: https://www.can-cia.org/can-knowledge.
8. DS3231. I2C-Integrated RTC/TCXO/Crystal. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ds3231.pdf.
9. ELM327. URL: https://www.elmelectronics.com/products/ics/obd/.
10. ESP32 Series. Datasheet Version5.2. URL: https://documentation.espressif.com/esp32_datasheet_en.pdf.
11. IEC 31010:2019. Risk management — Risk assessment techniques. URL: https://webstore.iec.ch/en/publication/59809/.
12. Introduction to Internal Combustion Engines. — Wiley. URL: https://catalogimages.wiley.com/images/db/pdf/9781118533314.excerpt.pdf.
13. IS0 9141-2. Road vehicles - Diagnostic systems Part 2: CARB requirements for interchange of digital information URL: https://cdn.standards.iteh.ai/samples/16738/20ff360c25c6462b811ebc0c9256eee0/ISO-9141-2-1994.pdf.
14. ISO 11898-1:2024. Road vehicles — CAN. Part 1: Data link layer and physical coding sublayer. URL: https://cdn.standards.iteh.ai/samples/86384/ca6e28c79b6b4388a5104dfda2332666/ISO-11898-1-2024.pdf.

15. ISO 11898-2:2024. Road vehicles — CAN. Part 2: High-speed PMA sublayer. URL: https://cdn.standards.iteh.ai/samples/85120/1e8cd4d7185d4fc9969cf8bcd4de5efc/ISO-11898-2-2024.pdf.
16. ISO 12100:2010. Safety of machinery — General principles for design — Risk assessment and risk reduction. URL: https://www.iso.org/standard/51528.html.
17. ISO 16750-3. Road vehicles — Part 3: Mechanical loads. URL: https://ru.scribd.com/document/944041310/ISO-16750-3-2023.
18. ISO 16750-4. Road vehicles — Part 4: Climatic loads. URL: https://u.dianyuan.com/bbs/u/44/1155638751.pdf.
19. ISO 9141:1989. Road vehicles. URL: https://www.iso.org/obp/ui/en/.
20. L9637. Monolithic bus driver with ISO 9141 interface. URL: https://www.st.com/resource/en/datasheet/l9637.pdf.
21. LM193, LM293 LM393, LM393W. Low-power, dual-voltage comparators. URL: https://datasheet4u.com/download/1495622/LM393.html.
22. MC33199. Automotive ISO 9141 Serial Link Driver URL: https://www.nxp.com/docs/en/fact-sheet/MC33199FS.pdf.
23. MCP2515 Stand-Alone CAN Controller with SPI Interface. Microchip Technology. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf.
24. Rajamani R. Vehicle Dynamics and Control. — Springer, 2012. URL: https://link.springer.com/book/10.1007/978-1-4614-1433-9.
25. Ross-Tech. VCDS: Windows-based Diagnostic Software for VW / Audi / Seat / Skoda. URL: https://www.ross-tech.com/vag-com/VCDS.php.
26. SJA1000 Stand-alone CAN controller. URL: https://www.alldatasheet.com/datasheet-pdf/download/19144/PHILIPS/SJA1000.html.
27. STM32F103C8T6 ARM Cortex-M3 Microcontroller Board. URL: https://handsontec.com/dataspecs/module/STM32F103%20module.pdf.
28. UMW LM2596/LM2596HV. 3A Step-Down Voltage Regulator. URL: https://images.tuyacn.com/smart/A_TUYA/cropper/LM2596S.pdf.

29. URB2405YMD-10WR3. DC/DC перетворювач. URL: https://kosmodrom.ua/dc-dc-peretvoryuvachi-na-drukovanu-platu/hlk-10d2405a.html.
30. Закон України “Про охорону праці”. URL: https://zakon.rada.gov.ua/laws/show/2694-12#Text.
31. Кодекс законів про працю України : Кодекс України від 10.12.1971 № 322-VIII. URL: https://zakon.rada.gov.ua/go/322-08.
32. Про затвердження Вимог безпеки та захисту здоров’я під час використання виробничого обладнання працівниками : Наказ Міністерства соціальної політики України від 28.12.2017 № 2072. URL: https://zakon.rada.gov.ua/go/z0097-18.
33. Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями : Наказ Міністерства соціальної політики України від 14.02.2018 № 207. URL: https://zakon.rada.gov.ua/go/z0508-18.
34. Про затвердження Правил безпечної експлуатації електроустановок споживачів : Наказ Міністерства праці та соціальної політики України від 09.01.1998 № 4. URL: https://zakon.rada.gov.ua/go/z0093-98.
35. Про затвердження Правил пожежної безпеки в Україні : Наказ МВС України від 30.12.2014 № 1417. URL: https://zakon.rada.gov.ua/go/z0252-15.
36. Про затвердження Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці : Наказ Держнаглядохоронпраці України від 26.01.2005 № 15. URL: https://zakon.rada.gov.ua/go/z0231-05.
37. Проєктування систем автоматизації : навч. посіб. — Київ : КПІ ім. Ігоря Сікорського, 2023. URL: https://ela.kpi.ua/server/api/core/bitstreams/b973e8da-7468-42a5-8085-12b287d85407/content.
38. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 : Постанова Головного державного санітарного лікаря України від 01.12.1999 № 42. URL: https://zakon.rada.gov.ua/go/va042282-99.

ДОДАТОК А
ТЕКСТ ГОЛОВНОЇ ПРОГРАМИ КОНТРОЛЕРА

```
#define WorkMode

#include "mcp2515.h"
#include <LiquidCrystal.h>
#include <EEPROM.h>

#define RSPin 8
#define ENPin 7
#define D4Pin A3
#define D5Pin A2
#define D6Pin A1
#define D7Pin A0
LiquidCrystal Lcd(RSPin, ENPin, D4Pin, D5Pin, D6Pin, D7Pin);

#define ButtonPin 4
#define BusserPin 5

#define MCPClock MCP_8MHZ
#define CANSpeed1 CAN_500KBPS
#define CCPin1 9
#define CAN_INT1 2
volatile bool canDataReceived1 = false;
MCP2515 mcp2515_1(CCPin1);
#define CANSpeed2 CAN_100KBPS
#define CCPin2 10
#define CAN_INT2 3
volatile bool canDataReceived2 = false;
MCP2515 mcp2515_2(CCPin2);
struct can_frame canMsg;

byte startCommunication[]                = {0x81, 0x10, 0xf1, 0x81, 0x03};
byte readDataByLocalIdentifier []        = {0x82, 0x10, 0xF1, 0x21, 0x01, 0xA5};
byte readDTCByStatus[]                   = {0x84, 0x10, 0xF1, 0x18, 0x00, 0x00, 0x00, 0x9D};
byte clearDiagnosticInformation[]        = {0x83, 0x10, 0xf1, 0x14, 0x00, 0x00, 0x98};
byte stopCommunication[]                 = {0x81, 0x10, 0xf1, 0x82, 0x04};

byte SymbGrad[] = {0x1c, 0x14, 0x1c, 0x03, 0x04, 0x04, 0x04, 0x03};
byte SymbElec[] =  {0x03, 0x06, 0x0c, 0x1f, 0x06, 0x0c, 0x18, 0x00};
byte SymbDros[] = {0x01, 0x02, 0x0c, 0x0e, 0x06, 0x08, 0x10, 0x00};
byte SymbBenz[] = {0x00, 0x1d, 0x12, 0x1f, 0x1b, 0x1b, 0x1f, 0x00};
byte SymbDvig[] = {0x1c, 0x08, 0x1c, 0x1d, 0x1f, 0x1d, 0x1c, 0x00};
byte SymbDist[] =  {0x04, 0x06, 0x07, 0x06, 0x04, 0x04, 0xe, 0x00};

#define SerialSpeed 115200

#define SIGNATURE 0xAA55AA55u
```

```
#define FRAME_INTERVALS_SIZE 100
#define SWAP_BYTES_UINT32(value)    ((((value) >> 24) & 0x000000FF) | (((value) >> 8) & 0x0000FF00) | (((value) << 8) & 0x00FF0000) | (((value) << 24) & 0xFF000000))

struct FrameTime
{
   uint32_t ID;
   uint32_t ms;
};
FrameTime frameTimes[FRAME_INTERVALS_SIZE] = {{ 0, 0 }};
uint16_t frameTimesAmount = 0;

struct OutCANFrame
{
   uint32_t signature = SWAP_BYTES_UINT32(SIGNATURE);
   uint32_t id;
   uint16_t interval;
   uint8_t  length;
   uint8_t  data[8];
};
OutCANFrame outCANFrame;

#define Shift 2
#define ErrorMaxCount 2
volatile word Errors[ErrorMaxCount + 1] = {0, 0, 0};

#define BUFFER_SIZE 100
volatile byte buffer[BUFFER_SIZE];
volatile byte BufCnt = 0;

#define PageWorkData      1
#define PageCounter1Data  2
#define PageCounter2Data  3
#define PageErrorData     4

volatile byte Page = PageWorkData;
volatile bool DataChange = false;
volatile unsigned long PredTime = 0;
volatile bool IsConnected = false;

struct Counter {
  float Probeg;
  float Fuel;
};
Counter Counters[4];

#define Pause 200
#define StartEEPROM 100
bool button  = false;
bool press_flag = false;
bool long_press_flag = false;
```

```
unsigned long last_press = 0;
volatile unsigned long LastQwery = 0;

void MCPInit() {
 int Filtr = 0x000;
 if (mcp2515_1.reset() == MCP2515::ERROR_OK) {
  mcp2515_1.setBitrate(CANSpeed1, MCPClock);
  mcp2515_1.setFilterMask(MCP2515::MASK0, false, (Filtr == 0x000) ? 0x000 : 0xFF0);
  mcp2515_1.setFilterMask(MCP2515::MASK1, false, (Filtr == 0x000) ? 0x000 : 0xFF0);
  mcp2515_1.setFilter(MCP2515::RXF0, false, Filtr);
  mcp2515_1.setFilter(MCP2515::RXF1, false, Filtr);
  mcp2515_1.setFilter(MCP2515::RXF2, false, Filtr);
  mcp2515_1.setFilter(MCP2515::RXF3, false, Filtr);
  mcp2515_1.setFilter(MCP2515::RXF4, false, Filtr);
  mcp2515_1.setFilter(MCP2515::RXF5, false, Filtr);
  mcp2515_1.setNormalMode();
 }
 if (mcp2515_2.reset() == MCP2515::ERROR_OK) {
  mcp2515_2.setBitrate(CANSpeed2, MCPClock);
  mcp2515_2.setFilterMask(MCP2515::MASK0, false, (Filtr == 0x000) ? 0x000 : 0xFF0);
  mcp2515_2.setFilterMask(MCP2515::MASK1, false, (Filtr == 0x000) ? 0x000 : 0xFF0);
  mcp2515_2.setFilter(MCP2515::RXF0, false, Filtr);
  mcp2515_2.setFilter(MCP2515::RXF1, false, Filtr);
  mcp2515_2.setFilter(MCP2515::RXF2, false, Filtr);
  mcp2515_2.setFilter(MCP2515::RXF3, false, Filtr);
  mcp2515_2.setFilter(MCP2515::RXF4, false, Filtr);
  mcp2515_2.setFilter(MCP2515::RXF5, false, Filtr);
  mcp2515_2.setNormalMode();
 }
}

void CANInterrupt1() {
  canDataReceived1 = true;    //enable Can1
}

void CANInterrupt2() {
  canDataReceived2 = true;    // enable Can2
}

void SerialPrint(can_frame Msg) {
  outCANFrame.id = Msg.can_id;
  outCANFrame.length = Msg.can_dlc;
  for(int i = 0; i<Msg.can_dlc; i++)
    outCANFrame.data[i] = Msg.data[i];
  uint32_t currentTime = millis();
  outCANFrame.interval = getAndSaveIdInterval(outCANFrame.id, currentTime);
  uint8_t dataLength = 11 + outCANFrame.length;
  Serial.write((uint8_t*)&outCANFrame, dataLength);
}

uint16_t getAndSaveIdInterval(uint32_t id, uint32_t currentTime)
```

```
{
  for (uint8_t index = 0; index < frameTimesAmount; index++) {
    if (frameTimes[index].ID == id) {
      uint32_t interval = currentTime - frameTimes[index].ms;
      frameTimes[index].ms = currentTime;
      return (interval < 65536) ? (uint16_t)interval : 0;
    }
  }
  if (frameTimesAmount < FRAME_INTERVALS_SIZE) {
    frameTimes[frameTimesAmount].ID = id;
    frameTimes[frameTimesAmount].ms = currentTime;
    frameTimesAmount++;
  }
  return 0;
}

void LcdPrint(can_frame Msg) {
 Lcd.clear();
 Lcd.setCursor(0, 0);
 Lcd.print(Msg.can_id,HEX); Lcd.print(" ");
 Lcd.print(Msg.can_dlc, HEX); Lcd.print(" ");
 for(int i = 0; i<3; i++) {
  if (Msg.data[i] < 0x10) Lcd.print("0");
  Lcd.print(Msg.data[i],HEX); Lcd.print(" ");
 }
 Lcd.setCursor(2, 1);
 for(int i = 3; i<Msg.can_dlc; i++) {
  if (Msg.data[i] < 16) Lcd.print("0");
  Lcd.print(Msg.data[i],HEX);
  Lcd.print(" ");
 }
}

void WriteEPROMData(byte OffMult = 1) {
 DataChange = false;
 int address = StartEEPROM * OffMult;
 EEPROM.write(address, 0);
 address += 1;
 for (byte i = 0; i < 3; i++) {
  EEPROM.put(address, Counters[i]);
  address += sizeof(Counter);
 }
 EEPROM.write(StartEEPROM * OffMult, 0x55);
 address += 1;
 if (OffMult == 1) WriteEPROMData(2);
}

void ReadEPROMData(byte OffMult = 1) {
 DataChange = false;
 int address = StartEEPROM * OffMult;
 if (EEPROM.read(address) == 0x55) {
```

```
  address += 1;
  for (byte i = 0; i < 3; i++) {
   EEPROM.get(address, Counters[i]);
   address += sizeof(Counter);
  }
 }
 else {
  if (OffMult == 1)
   ReadEPROMData(2);
  else {
   for (byte i = 0; i < 3; i++)
    Counters[i] = (Counter) {0,0};
  }
 }
 Counters[3] = {0, 0};
}

boolean DoCommand(byte *Command, byte CommandByteCount) {
 while ((millis() - LastQwery) < Pause) {};
 while(Serial.available() > 0) Serial.read();
 Serial.write(Command, CommandByteCount);
 Serial.flush();
 byte echoCount = 0;
 unsigned long startTime = millis();
 while (echoCount < CommandByteCount) {
  if (Serial.available() > 0) {
   Serial.read();
   echoCount++;
  }
  if (millis() - startTime > 50) break;
 }

 byte data_length = 0;
 BufCnt = 0;
 IsConnected = false;
 unsigned long Timer = millis();
 while ((millis() - Timer) < Pause) {
  while (Serial.available() > 0) {  // enable KLine
    byte curByte = Serial.read();
    buffer[BufCnt++] = curByte;
    if (BufCnt == 1) data_length = (curByte & 0x3F) + 3 + 1;
  }
  if (data_length > 0 && BufCnt == data_length) {
   byte crc = 0;
   for (byte i = 0; i < data_length-1; i++) crc = crc + buffer[i];
   IsConnected = (buffer[data_length-1] == crc ) && (buffer[3] == Command[3]+0x40);
   break;
  }
 }
 LastQwery = millis();
 return IsConnected;
```

```
}

void GetInfo(bool NeedGetErrors) {
 if (DoCommand(readDataByLocalIdentifier, sizeof(readDataByLocalIdentifier))) {
  if (PredTime == 0) PredTime = millis();
  else {
   unsigned long CurTime = millis();
   if (buffer[Shift + 15] > 50) {
    float Rashod    = (float) ((buffer[Shift + 30] << 8) + buffer[Shift + 29]) / 50;
    float DeltaTime    = (float) (CurTime - PredTime) / 1000;
    float DeltaRashod  = (float) Rashod * DeltaTime / 3600;
    float DeltaPath    = (float) buffer[Shift + 20] * 1000 * DeltaTime / 3600;
    int PredKM = (int) Counters[3].Probeg / 1000;
    for (byte i = 0; i <= 3; i++) {
     Counters[i].Probeg += DeltaPath;
     Counters[i].Fuel += DeltaRashod;
    }
    if ((int) Counters[3].Probeg / 1000 != PredKM)  WriteEPROMData();
    DataChange = true;
   }
   PredTime = CurTime;
  }
 }

 if (NeedGetErrors) {
  if (DoCommand(readDTCByStatus, sizeof(readDTCByStatus))) {
   byte errCnt = buffer[4];
    Errors[0] = errCnt;
    for (byte i = 1; i <= (errCnt > ErrorMaxCount ? ErrorMaxCount : errCnt); i++)
     Errors[i] = buffer[5 + ((i-1) * 3)] << 8 | buffer[6 + ((i-1) * 3)];
  }
 }
}

void PrintLeed(byte CelCount, int IntValue) {
 for (int i = 1; i < CelCount; i++) {
  long dd = 1;
  for (int k = 1; k <= (CelCount - i); k++) dd = dd * 10;
  if (IntValue < dd) Lcd.print(' ');
  else break;
 }
}

void printIntValue(byte Col, byte Row, int Value, byte PosCount, byte FinishChar = 32, bool
FinishBlank = false) {
 Lcd.setCursor(Col, Row);
 if (Value < 0) {
  Lcd.print('-');
  PosCount -= 1;
  Value = -Value;
 }
```

```
  PrintLeed(PosCount, Value);
  Lcd.print(Value, DEC);
  Lcd.write(byte(FinishChar));
  if (FinishBlank) Lcd.write(byte(32));
}

void printFloatValue(byte Col, byte Row, float Value, byte PosCount, byte ManCount, byte
FinishChar = 32, bool FinishBlank = false) {
  Lcd.setCursor(Col, Row);
  if (Value < 0) {
    Lcd.print('-');
    PosCount -= 1;
    Value = -Value;
  }
  PrintLeed(PosCount - ManCount - 1, (int) Value);
  Lcd.print(Value, ManCount);
  Lcd.write(byte(FinishChar));
  if (FinishBlank) Lcd.write(byte(32));
}

void printCounter(byte CNum, byte Row) {
  Lcd.setCursor(0, Row);
  Lcd.print(CNum + 1, DEC);
  Lcd.print(':');

  if (CNum == 0) printIntValue(2, Row, (unsigned long) Counters[CNum].Probeg / 1000, 7, 5, true);
  else if (CNum == 3) printFloatValue(2, Row, (float) Counters[CNum].Probeg / 1000, 7, 3, 5, true);
  else printFloatValue(2, Row, (float) Counters[CNum].Probeg / 1000, 7, 1, 5, true);
  float TekRash = (Counters[CNum].Probeg == 0) ? 0 : (Counters[CNum].Fuel * 100) /
(Counters[CNum].Probeg / 1000);
  printFloatValue(11, Row, TekRash, 4, 1, 3);
}

void printErrorText(int Err) {
  switch (Err) {
    case 0x0102:
    case 0x0103: Lcd.print("DMRV"); break;
    case 0x0117:
    case 0x0118: Lcd.print("Temp"); break;
    case 0x0122:
    case 0x0123: Lcd.print("DPDZ"); break;
    case 0x0325:
    case 0x0327:
    case 0x0328: Lcd.print("DDet"); break;
    case 0x0335: Lcd.print("DKV"); break;
    case 0x0501: Lcd.print("Speed"); break;
    case 0x0562:
    case 0x0563: Lcd.print("Voltage"); break;
    case 0x0601: Lcd.print("Immo"); break;
    case 0x1171:
    case 0x1172: Lcd.print("CO pot"); break;
```

```
    case 0x1509:
    case 0x1513:
    case 0x1514:
    case 0x0505: Lcd.print("RHH"); break;
    case 0x1602: Lcd.print("Volt off"); break;
    case 0x1612: Lcd.print("Reset err"); break;
    case 0x1620: Lcd.print("ROM err"); break;
    case 0x1621: Lcd.print("RAM err"); break;
    case 0x1622: Lcd.print("EEPROM err"); break;
    default: Lcd.print("Unknown"); break;
  }
}

void PrintErrors() {
  Lcd.clear();
  if (Errors[0] == 0) {
    Lcd.setCursor(0, 0);
    Lcd.print('O'); Lcd.print('K');
  }
  else {
    Lcd.setCursor(14, 0);
    Lcd.print(Errors[0], DEC);
    for (byte i = 1; i <= (Errors[0] > 2 ? 2 : Errors[0]); i++) {
      Lcd.setCursor(0, i - 1);
      Lcd.print('P');
      if (Errors[i] < 0x1000) Lcd.print('0');
      Lcd.print(Errors[i], HEX);
      Lcd.print(' ');
      printErrorText(Errors[i]);
    }
  }
}

void DisplayPage() {
  switch (Page) {
    case PageWorkData: {
      int CurPower = (int) ((buffer[Shift + 28] << 8) + buffer[Shift + 27]) / (6 * 5);
      CurPower = (CurPower > 100) ? 100 : CurPower;
      float P_Rash;
      if (buffer[Shift + 20] < 10)
        P_Rash = (float) ((buffer[Shift + 30] << 8) + buffer[Shift + 29]) / 50;
      else
        P_Rash = (float) ((buffer[Shift + 32] << 8) + buffer[Shift + 31]) / 128;
      if ((P_Rash > 100) || (P_Rash < 0)) P_Rash = 0;
      printIntValue  ( 0, 0, CurPower, 3, 4, true); // power
      printIntValue  ( 0, 1, (int) buffer[Shift + 13], 3, 2, true); // throttle
      printFloatValue( 5, 0, Counters[3].Probeg / 1000, 5, 1); // mileage
      printFloatValue( 5, 1, P_Rash, 4, 1, 3, true); // consumption
      printFloatValue(11, 0, (float) (buffer[Shift + 21]) / 20 + 5.2, 4, 1, 1); // voltage
      printIntValue  (11, 1, (int) buffer[Shift + 11] - 40, 4, 0); // temperature
      break;
```

```
    }
   case PageCounter1Data: {
      printCounter(0, 0);
      printCounter(1, 1);
      break;
    }
   case PageCounter2Data: {
      printCounter(2, 0);
      printCounter(3, 1);
      break;
    }
   case PageErrorData: {
      PrintErrors();
      break;
    }
  }
}

void ClearErrors() {
  if (DoCommand(clearDiagnosticInformation, sizeof(clearDiagnosticInformation)))
    Errors[0] = 0;
}

void ButtonTest() {
  button = !digitalRead(ButtonPin);
  if (button && !press_flag && (millis() - last_press) > 50) {
    press_flag = !press_flag;
    last_press = millis();
  }
  if (button && press_flag && (millis() - last_press) > 1000) {
    long_press_flag = !long_press_flag;
    last_press = millis();
    switch (Page) {
      case PageWorkData:
      case PageErrorData: {
        ClearErrors();
        break;
       }
      case PageCounter1Data:
      case PageCounter2Data: {
        Counters[(Page == PageCounter1Data) ? 1 : 2] = (Counter){0,0};
        break;
       }
    }
    WriteEPROMData();
  }
  if (!button && press_flag && long_press_flag) {
    press_flag = !press_flag;
    long_press_flag = !long_press_flag;
  }
  if (!button && press_flag && !long_press_flag) {
```

```
    press_flag = !press_flag;
    Page = (Page == PageErrorData) ? PageWorkData : Page + 1;
    Lcd.clear();
  }
}
void InitConnect() {
  DoCommand(stopCommunication, sizeof(startCommunication));
  DoCommand(startCommunication, sizeof(startCommunication));
}

void setup() {
  Lcd.begin(16, 2);
  Lcd.createChar(0, SymbGrad);
  Lcd.createChar(1, SymbElec);
  Lcd.createChar(2, SymbDros);
  Lcd.createChar(3, SymbBenz);
  Lcd.createChar(4, SymbDvig);
  Lcd.createChar(5, SymbDist);
  #ifdef WorkMode
    Serial.begin(10400);
  #else
    Lcd.clear();
    Lcd.setCursor(0, 0);
    Lcd.print("--- CAN Read ---");
    Lcd.setCursor(0, 1);
    Lcd.print("ID  DLC   DATA");
    Serial.begin(SerialSpeed);
  #endif
  MCPInit();
  pinMode(CAN_INT1, INPUT);
  pinMode(CAN_INT2, INPUT);
  attachInterrupt(digitalPinToInterrupt(CAN_INT1), CANInterrupt1, FALLING);
  attachInterrupt(digitalPinToInterrupt(CAN_INT2), CANInterrupt2, FALLING);
  pinMode(ButtonPin, INPUT_PULLUP);
  pinMode(BusserPin, OUTPUT);
  ReadEPROMData();
  #ifdef WorkMode
    byte Cnt = 9;
    while (!IsConnected) {
      if (Cnt == 9) {
        Lcd.clear(); Lcd.setCursor(0, 0);
        Lcd.print("Connect"); Cnt = 0;
      }
      Lcd.print("."); Cnt++; InitConnect();
    }
    Lcd.setCursor(0, 1); Lcd.print("OK");
    delay(250);
    Lcd.clear();
  #endif
}
```

```
void loop() {
  if (canDataReceived1 || canDataReceived2){
    MCP2515::ERROR Res;
    if (canDataReceived1) {
      canDataReceived1 = false;
      Res = mcp2515_1.readMessage(&canMsg);
    }
    else {
      canDataReceived2 = false;
      Res = mcp2515_2.readMessage(&canMsg);
    }
    if (Res == MCP2515::ERROR_OK) {
      #ifndef WorkMode
        SerialPrint(canMsg);
        LcdPrint(canMsg);
      #else
        SaveCanMsg(canMsg);
      #endif
    }
  }
  #ifdef WorkMode
    ButtonTest();
    if (!IsConnected)
      InitConnect();
    if (IsConnected) {
      GetInfo(Page == PageErrorData);
      DisplayPage();
    }
  #endif
}
```

ДОДАТОК Б

ТЕКСТ ПРОГРАМИ ЕМУЛЯТОРА МОДУЛЯ КОНТРОЛЕРУ MCP2515

```
#define ChipName "CAN-1-chip"
#define TimerInterval 10
#define ChipNmb 0x1
#define PackNmb 0xD0
#include "wokwi-api.h"
#include <stdio.h>
#include <stdlib.h>
uint8_t cur_data_ind = 0;
uint8_t cur_data_pos = 0;
#define MAX_BUF_SIZE 20

typedef struct {
  pin_t pin_int;
  pin_t pin_cs;
  uint32_t spi;
  uint8_t spi_buffer[1];
  uint8_t tx_buffer[MAX_BUF_SIZE];
  uint32_t byte_count;
  bool session_active;
} chip_state_t;

static void chip_timer_event(void *user_data);
static void chip_pin_change(void *user_data, pin_t pin, uint32_t value);
static void chip_spi_done(void *user_data, uint8_t *buffer, uint32_t count);

void chip_init(void) {
  chip_state_t *chip = malloc(sizeof(chip_state_t));
  chip->byte_count = 0;
  chip->session_active = false;
  chip->pin_cs = pin_init("CS", INPUT_PULLUP);
  chip->pin_int = pin_init("INT", OUTPUT_LOW);
  const pin_watch_config_t watch_config = {
    .edge = BOTH,
    .pin_change = chip_pin_change,
    .user_data = chip,
  };
  pin_watch(chip->pin_cs, &watch_config);
  const spi_config_t spi_config = {
    .sck = pin_init("SCK", INPUT),
    .miso = pin_init("MISO", INPUT),
    .mosi = pin_init("MOSI", INPUT),
    .done = chip_spi_done,
    .user_data = chip,
  };
  chip->spi = spi_init(&spi_config);
  const timer_config_t timer_config = {
```

```
    .callback = chip_timer_event,
    .user_data = chip,
  };
  timer_t timer_id = timer_init(&timer_config);
  timer_start(timer_id, TimerInterval * 100000, true);
}

void chip_pin_change(void *user_data, pin_t pin, uint32_t value) {
  chip_state_t *chip = (chip_state_t*)user_data;
  if (pin == chip->pin_cs) {
    if (value == LOW) {
      chip->byte_count = 0;
      for(uint8_t i = 0; i<MAX_BUF_SIZE; i++) chip->tx_buffer[i] = 0x00;
      chip->session_active = true;
      spi_start(chip->spi, &chip->tx_buffer[chip->byte_count], 1);
    }
    else {
      chip->session_active = false;
      spi_stop(chip->spi);
    }
  }
}

static uint8_t NextByte() {
  uint8_t res = 0;
  switch (cur_data_pos){
    case 0: res = ChipNmb;
          break;
    case 1: res = PackNmb + cur_data_ind;
          break;
    case 2: res = 8;
          break;
    default:
          res = rand() % (256);
          break;
  }
  cur_data_pos = (cur_data_pos == 10) ? 0 : cur_data_pos + 1;
  if (cur_data_pos == 0) cur_data_ind = (cur_data_ind == 9) ? 0 : cur_data_ind + 1;
  return res;
}

void prepare_response(chip_state_t *chip) {
  switch (chip->byte_count){
    case 1:
      switch (chip->tx_buffer[0]){
        case 0xC0: //INSTRUCTION_RESET
          break;
        case 0xA0:
          chip->tx_buffer[chip->byte_count] = 0x01;
          break;
      }
```

```
    break;
   case 2:
    switch (chip->tx_buffer[0]){
     case 0x02: //INSTRUCTION_WRITE
      break;
     case 0x03: //INSTRUCTION_READ
      switch (chip->tx_buffer[1]) {
       case 0x0E:
        chip->tx_buffer[chip->byte_count] = 0x80;
        break;
       case 0x61: {
        uint32_t id = (uint32_t)(NextByte() << 8) + NextByte();
        chip->tx_buffer[chip->byte_count] = (uint8_t)(id >> 3);
        chip->tx_buffer[chip->byte_count+1] = (uint8_t)((id & 0x07) << 5);
        chip->tx_buffer[chip->byte_count+2] = 0x00;
        chip->tx_buffer[chip->byte_count+3] = 0x00;
        chip->tx_buffer[chip->byte_count+4] = NextByte();
        break;}
       case 0x66:
        for (uint8_t i = 0; i<8; i++)
         chip->tx_buffer[chip->byte_count+i] = NextByte();
        break;
      }
    }
    break;
   case 3:
    break;
  }
}

void chip_spi_done(void *user_data, uint8_t *buffer, uint32_t count) {
  chip_state_t *chip = (chip_state_t*)user_data;
  if (!chip->session_active || count == 0)
    return;
  if (chip->byte_count < MAX_BUF_SIZE)
    chip->byte_count++;
  prepare_response(chip);
  if (chip->byte_count < MAX_BUF_SIZE)
    spi_start(chip->spi, &chip->tx_buffer[chip->byte_count], 1);
  return;
}

void chip_timer_event(void *user_data) {
  chip_state_t *chip = (chip_state_t*)user_data;
  pin_write(chip->pin_int, HIGH);
  pin_write(chip->pin_int, LOW);
}
```

ДОДАТОК В

ТЕКСТ ПРОГРАМИ ЕМУЛЯТОРА МОДУЛЯ КОНТРОЛЕРУ K-LINE

```
#define ChipName "K-Line-chip "
#include "wokwi-api.h"
#include <stdio.h>
#include <stdlib.h>

#define TimerInterval 300
#define HEADER_SIZE 3 // Format, Target, Source

#define   CMD_STC    0x81  //startCommunication
#define   CMD_SPC    0x82  //stopCommunication
#define   CMD_STDS   0x10  //startDiagnosticSession
#define   CMD_SPDS   0x20  //stopDiagnosticSession
#define   CMD_ER     0x11  //ecuReset
#define   CMD_CDI    0x14  //clearDiagnosticInformation
#define   CMD_RDTCBS 0x18  //readDiagnosticTroubleCodesByStatus
#define   CMD_REI    0x1A  //readEcuIdentification
#define   CMD_RDBLI  0x21  //readDataByLocalIdentifier
#define   CMD_RMBA   0x23  //readMemoryByAddress
#define   CMD_IOCBLI 0x30  //inputOutputControlByLocalIdentifier
#define   CMD_WDBLI  0x3B  //writeDataByLocalIdentifier
#define   CMD_TP     0x3E  //testerPresent

typedef struct {
  uart_dev_t uart;
  uint8_t buffer[250];
  uint8_t rx_index;
  uint8_t expected_packet_length;
  bool IsErrosExsist;
} chip_state_t;

static void chip_timer_event(void *user_data);

static void on_uart_rx_data(void *user_data, uint8_t byte);
static void on_uart_write_done(void *user_data);

void chip_init(void) {
  chip_state_t *chip = malloc(sizeof(chip_state_t));

  const uart_config_t uart_config = {
    .tx = pin_init("TX", INPUT_PULLUP),
    .rx = pin_init("RX", INPUT),
    .baud_rate = 10400,
    .rx_data = on_uart_rx_data,
    .write_done = on_uart_write_done,
    .user_data = chip,
  };
```

```
  chip->uart = uart_init(&uart_config);
  chip->IsErrosExsist = false;

  const timer_config_t timer_config = {
    .callback = chip_timer_event,
    .user_data = chip,
  };
  timer_t timer_id = timer_init(&timer_config);
  timer_start(timer_id, TimerInterval * 100000, true);
}

static void process_kline_packet(chip_state_t *chip) {
  if (chip->rx_index < 5)  // Incorrect size
    return;
  uint8_t reqCount = chip->rx_index;
  uint8_t crc = 0;
  for (uint8_t i = 0; i < chip->rx_index-1; i++)
    crc = crc + chip->buffer[i];
  if (chip->buffer[chip->rx_index-1] != crc) //Incorrect CRC
    return;

  chip->buffer[chip->rx_index++] = 0x80;
  chip->buffer[chip->rx_index++] = chip->buffer[2];
  chip->buffer[chip->rx_index++] = chip->buffer[1];
  chip->buffer[chip->rx_index++] = chip->buffer[3] + 0x40;

  switch (chip->buffer[3]){
    case CMD_STC: //startCommunication
      chip->buffer[chip->rx_index++] = 0x6B;
      chip->buffer[chip->rx_index++] = 0x8F;
      break;
    case CMD_SPC: //stopCommunication
      break;
    case CMD_RDBLI: //readDataByLocalIdentifier
      chip->buffer[chip->rx_index++] = chip->buffer[4];
      for (uint8_t i = 0; i < 8; i++)
        chip->buffer[chip->rx_index++] = rand() % (256);
      chip->buffer[chip->rx_index++] = (70 + rand() % 10) + 40;
      chip->buffer[chip->rx_index++] = rand() % (256);
      chip->buffer[chip->rx_index++] = (20 + rand() % 10);
      chip->buffer[chip->rx_index++] = (1500 + rand() % 10) / 40;
      chip->buffer[chip->rx_index++] = (1500 + rand() % 10) / 10;
      for (uint8_t i = 0; i < 4; i++)
        chip->buffer[chip->rx_index++] = rand() % (256);
      chip->buffer[chip->rx_index++] = (100 + rand() % 10);
      chip->buffer[chip->rx_index++] = (170 + rand() % 10);
      for (uint8_t i = 0; i < 5; i++)
        chip->buffer[chip->rx_index++] = rand() % (256);
      chip->buffer[chip->rx_index++] = (2 + rand() % 10)*125;
      chip->buffer[chip->rx_index++] = 5;
      chip->buffer[chip->rx_index++] = (8 + rand() % 1)*50;
```

```
    chip->buffer[chip->rx_index++] = 1;
    chip->buffer[chip->rx_index++] = (1 + rand() % 30)*128;
    chip->buffer[chip->rx_index++] = 4;
    for (uint8_t i = 0; i < 10; i++)
      chip->buffer[chip->rx_index++] = rand() % (256);
    break;
  case CMD_RDTCBS: //readDiagnosticTroubleCodesByStatus
    if (chip->IsErrosExsist) {
      chip->buffer[chip->rx_index++] = 0x03;
      chip->buffer[chip->rx_index++] = 0x01;
      chip->buffer[chip->rx_index++] = 0x18;
      chip->buffer[chip->rx_index++] = 0xE0;
      chip->buffer[chip->rx_index++] = 0x03;
      chip->buffer[chip->rx_index++] = 0x35;
      chip->buffer[chip->rx_index++] = 0xE0;
      chip->buffer[chip->rx_index++] = 0x16;
      chip->buffer[chip->rx_index++] = 0x02;
      chip->buffer[chip->rx_index++] = 0xE0;
    }
    else {
      chip->buffer[chip->rx_index++] = 0x00;
    }
    break;
  case CMD_CDI: //clearDiagnosticInformation
    chip->buffer[chip->rx_index++] = chip->buffer[4];
    chip->buffer[chip->rx_index++] = chip->buffer[5];
    chip->IsErrosExsist = false;
    break;
  case CMD_STDS: //startDiagnosticSession
    chip->buffer[chip->rx_index++] = 0x81;
    break;
  case CMD_SPDS: //stopDiagnosticSession
    break;
  case CMD_TP:  //testerPresent
    break;
  case CMD_ER:  //ecuReset
    break;
  case CMD_REI:  //readEcuIdentification
    chip->buffer[chip->rx_index++] = chip->buffer[4];
    chip->buffer[chip->rx_index++] = 'E';
    chip->buffer[chip->rx_index++] = 'M';
    chip->buffer[chip->rx_index++] = 'U';
    chip->buffer[chip->rx_index++] = 'L';
    break;
}
chip->buffer[reqCount] += (chip->rx_index-3 - reqCount);

crc = 0;
for (uint8_t i = reqCount; i < chip->rx_index; i++)
  crc = crc + chip->buffer[i];              // CRC
chip->buffer[chip->rx_index++] = crc;
```

```
  uart_write(chip->uart, chip->buffer, chip->rx_index);
}

static void on_uart_rx_data(void *user_data, uint8_t byte) {
  chip_state_t *chip = (chip_state_t*)user_data;

  chip->buffer[chip->rx_index++] = byte;
  if (chip->rx_index == 1) {
      uint8_t data_length = byte & 0x3F;
      if (data_length > 0) {
         chip->expected_packet_length = HEADER_SIZE + data_length + 1;
      } else {
         chip->expected_packet_length = 0;
         chip->rx_index = 0;
      }
    }
    if (chip->expected_packet_length > 0 && chip->rx_index == chip->expected_packet_length) {
      process_kline_packet(chip);
      chip->rx_index = 0;
      chip->expected_packet_length = 0;
    }
}

static void on_uart_write_done(void *user_data) {
  chip_state_t *chip = (chip_state_t*)user_data;
}

void chip_timer_event(void *user_data) {
  chip_state_t *chip = (chip_state_t*)user_data;
  chip->IsErrosExsist = true;
}
```